

Universidade do Minho

Departamento de Matemática

# **COMPLEMENTOS DE ANÁLISE NUMÉRICA**

## **Valores e Vectores Próprios**

M. J. Soares, M. I. Falcão

2000/2001

# Conteúdo

<b>Introdução</b>	<b>1</b>
<b>1 Resultados Básicos</b>	<b>2</b>
1.1 Valores e vectores próprios . . . . .	2
1.2 Transformações de semelhança . . . . .	5
1.3 Forma canónica de Jordan . . . . .	8
1.4 Propriedades de matrizes hermitianas . . . . .	9
1.5 Estimativas “a priori” para valores próprios . . . . .	10
1.6 Breve referência a problemas de condicionamento . . . . .	13
1.7 Notas e referências adicionais . . . . .	15
1.8 Exercícios . . . . .	15
<b>2 Matrizes de Transformação</b>	<b>22</b>
2.1 Matrizes elementares . . . . .	23
2.2 Utilização de matrizes elementares na redução de um vector coluna . . . . .	25
<b>3 Métodos Iterativos</b>	<b>34</b>
3.1 Método da potência . . . . .	34
3.2 Aceleração de convergência . . . . .	38
3.3 Deflação . . . . .	40
3.4 Método da potência inverso . . . . .	41

3.5	Notas e referências adicionais . . . . .	44
3.6	Exercícios . . . . .	45
<b>4</b>	<b>Métodos de Transformação</b>	<b>48</b>
4.1	Método de Jacobi . . . . .	48
4.2	Determinação de valores próprios de matrizes tridiagonais – método da sequência de Sturm . . . . .	52
4.3	Método QR Básico . . . . .	57
4.4	Método QR com translação de origem . . . . .	62
4.5	Método QR duplo . . . . .	65
4.6	Notas e referências adicionais . . . . .	68
4.7	Exercícios . . . . .	69

# Introdução

A primeira parte do curso de Complementos de Análise Numérica é dedicada ao estudo de métodos numéricos para a resolução do problema da determinação dos valores e vectores próprios de uma matriz quadrada. Problemas deste tipo ocorrem, por exemplo, em Física e Engenharia, associados ao estudo de problemas de vibração. Também, em Estatística, a análise de variâncias, passa pela resolução de um problema de valores próprios. A determinação do número de condição de uma matriz, utilizando a norma espectral (importante, como sabemos, no estudo do condicionamento da resolução de sistemas de equações lineares) exige também o conhecimento dos valores próprios de maior e menor módulo de uma certa matriz. De modo semelhante, o conhecimento de valores próprios de certas matrizes é importante para a determinação das condições de estabilidade de métodos numéricos de resolução de sistemas de equações diferenciais (ordinárias ou de derivadas parciais). Estes são apenas alguns exemplos, de áreas de aplicação bastante distintas, que ilustram a importância deste problema.

Apesar de possuir uma formulação matemática extremamente simples, o problema clássico de valores e vectores próprios, isto é, o da determinação de soluções não-triviais da equação  $Ax = \lambda x$ , pode considerar-se como um dos problemas básicos mais interessantes da Análise Numérica. Com efeito, na prática, ele origina uma variada gama de problemas cuja resolução de forma eficiente exige uma série de diferentes algoritmos. É precisamente do estudo de alguns desses algoritmos que nos iremos ocupar na primeira parte deste curso, recorrendo sempre que possível à sua implementação em MATLAB.

Estas notas estão divididas em quatro capítulos: *Resultados básicos* - onde são apresentados/relembrados alguns dos resultados mais importantes da teoria de valores e vectores próprios; *Matrizes de transformação* - onde são definidas as vulgarmente chamadas matrizes elementares que desempenham um papel importante na redução, por semelhança, de uma matriz  $A$  a uma forma “mais” simples; *Métodos iterativos* e *Métodos de transformação*. Nestes dois últimos capítulos são descritos vários algoritmos usados para o cálculo de valores e vectores próprios de uma dada matriz. No final de cada capítulo são propostos vários problemas para acompanhamento das aulas teórico-práticas. Os exercícios cuja resolução deve ser feita recorrendo ao MATLAB estão devidamente assinalados. São também apresentados, em anexo, alguns projectos envolvendo a escrita de programas em MATLAB, com o objectivo de ilustrar determinados resultados e/ou aplicá-los a matrizes de maior dimensão.

# 1. Resultados Básicos

Neste capítulo, faremos uma breve revisão da teoria básica de valores e vectores próprios. A sua finalidade é, essencialmente, a de relembrar alguns resultados particularmente relevantes para este curso, bem como permitir uma uniformização de notações.

## 1.1 Valores e vectores próprios

Em tudo quanto se segue, sendo  $K$  um corpo, denotaremos por  $M_{m,n}(K)$  o conjunto das matrizes do tipo  $m \times n$  sobre  $K$  e por  $M_n(K)$  o conjunto das matrizes quadradas de ordem  $n$  de elementos em  $K$ . Em geral,  $K = \mathbb{C}$ , e nesse caso simplificaremos as notações para  $M_{m,n}$  e  $M_n$ , respectivamente.

**Definição 1.1** *Sejam  $A \in M_n$ ,  $\lambda \in \mathbb{C}$  e  $x \in \mathbb{C}^n$ ,  $x \neq 0$ . Diz-se que  $x$  é um **vector próprio** de  $A$  associado ao valor próprio  $\lambda$  se e só se*

$$Ax = \lambda x. \quad (1.1)$$

A equação (1.1) pode ser reescrita como

$$(A - \lambda I)x = 0, \quad (1.2)$$

que é a forma matricial de um sistema homogéneo de  $n$  equações lineares. Este sistema tem uma solução não trivial,  $x \neq 0$ , se e só se a matriz correspondente for singular, i.e., se tivermos

$$\det(A - \lambda I) = 0. \quad (1.3)$$

Expandindo o determinante no lado esquerdo da equação (1.3), obtém-se uma equação polinomial explícita da forma

$$(-1)^n \lambda^n + \sum_{k=0}^{n-1} a_k \lambda^k = 0. \quad (1.4)$$

A equação (1.4) é chamada *equação característica* da matriz  $A$  e o polinómio no lado esquerdo desta equação é chamado *polinómio característico* de  $A$ . Como o coeficiente de  $\lambda^n$  é não-nulo, segue-se que a equação (1.4) tem  $n$  raízes (contando multiplicidades), ou seja que  $A$  tem  $n$  valores próprios. Sejam  $\lambda_1, \lambda_2, \dots, \lambda_n$  os valores próprios de  $A$ . Conclui-se imediatamente de (1.4) que

$$\det(A - \lambda I) = \prod_{i=1}^n (\lambda_i - \lambda).$$

Tem-se assim a seguinte identidade

$$\det A = \prod_{i=1}^n \lambda_i. \quad (1.5)$$

Associado a cada valor próprio  $\lambda_k$  existe, pelo menos, um vector próprio  $x_k$ , solução de

$$(A - \lambda_k I)x = 0.$$

Note-se que, sendo  $x_k$  um vector próprio, também  $\alpha x_k$  ( $\alpha \neq 0$ ) é um vector próprio associado a  $\lambda_k$ . É por vezes conveniente escolher o escalar  $\alpha$  de modo que  $y_k = \alpha x_k$  seja *normalizado*, isto é, possua uma determinada propriedade. Em geral, pretende-se que

$$\|y_k\|_2 = \{y_k^* y_k\}^{1/2} = 1.$$

(Nota: Dada uma certa matriz  $A$ , usaremos a notação  $A^*$  para designar a matriz transconjugada de  $A$ , i.e.,  $A^* = \overline{A^T}$ .)

**Definição 1.2** *O conjunto dos valores próprios de  $A$  chama-se **espectro** de  $A$  e denota-se por  $\sigma(A)$ . Ao número*

$$\rho(A) = \max\{|\lambda| : \lambda \in \sigma(A)\} \quad (1.6)$$

*chamamos **raio espectral** de  $A$ .*

Seja  $g(t) = p(t)/q(t)$  uma função racional. Então,  $g(A) = p(A)[q(A)]^{-1}$  está definido desde que  $q(A)$  seja não-singular. Facilmente se verifica que, sendo  $\lambda$  um valor próprio de  $A$ , então  $g(\lambda)$  é um valor próprio de  $g(A)$ . Por exemplo, tem-se:

- $\lambda^2$  é um valor próprio de  $A^2$  (potência).
- $\lambda - h$  é um valor próprio de  $A - hI$  (translação).
- $1/\lambda$  é um valor próprio de  $A^{-1}$  (inversão).

Como  $\det(A - \lambda I) = \det(A - \lambda I)^T = \det(A^T - \lambda I)$ , conclui-se que  $A$  e  $A^T$  têm os mesmos valores próprios. Contudo, os vectores próprios de  $A$  e  $A^T$  são, em geral, diferentes.

**Definição 1.3** Se  $y \neq 0$  satisfaz

$$A^T y = \lambda y, \quad (1.7)$$

isto é, se  $y$  é um vector próprio de  $A^T$  associado ao valor próprio  $\lambda$ , diz-se que  $y$  é um **vector próprio à esquerda** de  $A$ .

(Tal designação justifica-se, uma vez que, de (1.7), se segue que  $y^T A = \lambda y^T$ .) Por vezes, para evitar ambiguidades, referir-nos-emos aos vectores próprios de  $A$  definidos por (1.1) como vectores próprios à direita de  $A$ . Quando, pelo contexto, não seja necessária distinção, diremos apenas vectores próprios.

**Teorema 1.1 (Princípio de biortogonalidade)** *Sejam  $A$  uma matriz quadrada e  $\lambda, \mu \in \sigma(A)$  com  $\lambda \neq \mu$ . Se  $x$  é um vector próprio à direita de  $A$  associado a  $\lambda$  e  $y$  é um vector próprio à esquerda de  $A$  associado a  $\mu$ , então*

$$y^T x = 0. \quad (1.8)$$

**Dem.** Temos

$$y^T Ax = y^T (\lambda x) = \lambda (y^T x)$$

e

$$y^T Ax = (\mu y^T)x = \mu (y^T x),$$

donde se segue imediatamente que

$$(\lambda - \mu)y^T x = 0.$$

A relação (1.8) deduz-se de imediato, atendendo a que  $\lambda \neq \mu$ . □

**Teorema 1.2** *Vectores próprios associados a valores próprios distintos são linearmente independentes. Mais precisamente, se  $x_1, x_2, \dots, x_r$  são vectores próprios associados a valores próprios  $\lambda_1, \lambda_2, \dots, \lambda_r$  de uma certa matriz  $A$ , e se  $\lambda_i \neq \lambda_j$  ( $i \neq j; i, j = 1, \dots, r$ ), então  $x_1, x_2, \dots, x_r$  são linearmente independentes.*

**Dem.** Veja, por exemplo, [Hor85, pp 47,48]. □

**Teorema 1.3** *Seja  $\lambda$  um valor próprio de uma certa matriz  $A$ , de multiplicidade  $m$  (i.e.,  $\lambda$  é um zero de ordem  $m$  do polinómio característico de  $A$ ). Então, existem no máximo  $m$  vectores próprios linearmente independentes associados a esse valor próprio.*

**Dem:** Veja, por exemplo, [Hor85, p.61]. □

**Nota 1.1** Ao número máximo de vectores próprios linearmente independentes associados ao valor próprio  $\lambda$ , que não é mais do que a dimensão do subespaço vectorial de  $\mathbb{C}^n$

$$U_\lambda = \{x \in \mathbb{C}^n : Ax = \lambda x\},$$

chamamos multiplicidade *geométrica* de  $\lambda$ . À multiplicidade de  $\lambda$  como raiz da equação característica chamamos multiplicidade *algébrica* de  $\lambda$ . O que este teorema afirma é, portanto, que a multiplicidade geométrica de um valor próprio não excede a sua multiplicidade algébrica. No que se segue, quando falarmos de multiplicidade de um valor próprio, sem especificar de que tipo de multiplicidade se trata, queremos referir-nos à multiplicidade algébrica.

**Definição 1.4** Uma matriz quadrada  $A$  de ordem  $n$  diz-se **não-deficiente** se possuir  $n$  vectores próprios linearmente independentes, dizendo-se **deficiente** no caso contrário.

O Teorema 1.2 mostra que, se  $A$  possui  $n$  valores próprios distintos, então  $A$  é não-deficiente. Se  $A$  possui valores próprios múltiplos,  $A$  pode ou não ser deficiente.

**Exemplo 1.1** A matriz

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

é não-deficiente, e a matriz

$$\begin{pmatrix} 3 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

(que tem também  $\lambda = 3$  como valor próprio duplo) é deficiente.

## 1.2 Transformações de semelhança

**Definição 1.5** Uma matriz  $B \in M_n$  diz-se **semelhante** a uma matriz  $A \in M_n$  e escreve-se  $B \sim A$  sse existe uma matriz invertível  $S \in M_n$  tal que

$$B = S^{-1}AS.$$

A transformação  $S \mapsto S^{-1}AS$  é chamada uma *transformação de semelhança* (por meio de  $S$ ). Em particular, se  $S$  é uma matriz unitária (i.e.  $S^{-1} = S^*$ ), dizemos tratar-se de uma transformação de semelhança *unitária*. De modo análogo, se  $S$  é uma matriz ortogonal (i.e.  $S^{-1} = S^T$ ), a



correspondente transformação diz-se *ortogonal*. Facilmente se verifica que a *semelhança* é uma relação de equivalência, fazendo assim sentido dizer-se que  $A$  e  $B$  são semelhantes.

**Teorema 1.4** *Matrizes semelhantes têm os mesmos valores próprios.*

**Dem.** Seja  $B = S^{-1}AS$  e suponhamos que  $\lambda$  é um valor próprio de  $A$ , ou seja, que

$$Ax = \lambda x \quad (x \neq 0). \quad (1.9)$$

Segue-se imediatamente de (1.9) que

$$S^{-1}ASS^{-1}x = \lambda S^{-1}x,$$

ou seja, temos

$$Bz = S^{-1}ASz = \lambda z,$$

onde

$$z = S^{-1}x.$$

Concluimos assim que  $\lambda$  é também valor próprio de  $B$ . Quanto aos vectores próprios  $x$  de  $A$  e  $z$  de  $S^{-1}AS$ , associados ao mesmo valor próprio tem-se que

$$x = Sz.$$

□

**Definição 1.6** *Diz-se que uma matriz  $A \in M_n$  é **diagonalizável** sse  $A$  é semelhante a uma matriz diagonal.*

**Teorema 1.5**  *$A \in M_n$  é diagonalizável sse é não-deficiente.*

**Dem.** Suponhamos que  $A$  é não-deficiente, isto é, que  $A$  possui  $n$  vectores próprios linearmente independentes  $x_i$ ;  $i = 1, \dots, n$ , associados aos valores próprios  $\lambda_i$ ;  $i = 1, \dots, n$  (não necessariamente distintos). Então

$$Ax_i = \lambda_i x_i; \quad i = 1, \dots, n.$$

Seja

$$X = (x_1 | \dots | x_n).$$

Tem-se então,

$$\begin{aligned} AX &= (Ax_1 | \dots | Ax_n) \\ &= (\lambda_1 x_1 | \dots | \lambda_n x_n) \\ &= (x_1 | \dots | x_n) \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \\ &= XD, \end{aligned}$$

ou seja, tem-se

$$X^{-1}AX = D,$$

onde  $D = \text{diag}(\lambda_i)$ . Note-se que  $X$  é invertível, porque as suas colunas são, por hipótese, linearmente independentes.

Reciprocamente, se existe uma matriz não-singular  $X$  tal que  $X^{-1}AX = D$  com  $D = \text{diag}(d_i)$  uma matriz diagonal tem-se

$$AX = XD,$$

ou seja, tem-se

$$(Ax_1 | \dots | Ax_n) = (d_1x_1 | \dots | d_nx_n).$$

Isto mostra que

$$Ax_i = d_ix_i,$$

ou seja, que os elementos da diagonal de  $D$  são valores próprios de  $A$  tendo como vectores próprios associados as colunas de  $X$ , as quais são linearmente independentes, uma vez que  $X$  é invertível.  $\square$

**Teorema 1.6 (Teorema de Schür)** *Dada uma matriz  $A \in M_n$ , existe uma matriz unitária  $P$  tal que  $P^*AP$  é uma matriz triangular superior. Por outras palavras, toda a matriz quadrada é unitariamente semelhante a uma matriz triangular superior. Além disso, se  $A \in M_n(\mathbb{R})$  e se todos os valores próprios de  $A$  são reais, então  $P$  pode ser escolhida como real e ortogonal.*

**Dem.** Veja, por exemplo, [Hor85, pp 79,80].  $\square$

**Nota 1.2** Uma versão mais geral do Teorema de Schür é a seguinte:

*Se  $A \in M_n(\mathbb{R})$ , então existe uma matriz real e ortogonal  $P$  tal que  $P^TAP$  tem a forma quasi-triangular; isto é, pode ser escrita na forma particionada*

$$P^TAP = \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1r} \\ & T_{22} & \dots & T_{2r} \\ & & \ddots & \\ & & & T_{rr} \end{pmatrix}$$

*onde cada bloco  $T_{ii}$  é uma matriz quadrada cuja ordem é, no máximo, dois; além disso,  $P$  pode ser escolhida por forma que os blocos  $T_{ii}$  de ordem 2 tenham apenas valores próprios complexos (que serão, evidentemente, conjugados); a demonstração deste resultado pode ser vista, por exemplo, em [Ste73, p.285].*

O Teorema 1.5 mostra que, em geral, uma matriz não pode ser reduzida à forma diagonal através de transformações de semelhança. A forma mais *compacta* que tais transformações produzem é descrita na secção seguinte.

### 1.3 Forma canónica de Jordan

**Definição 1.7** Uma matriz quadrada  $r \times r$  da forma

$$J_r(\lambda) = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & \lambda \end{pmatrix}$$

é chamada **um bloco de Jordan de ordem  $r$** .

Facilmente se verifica que  $J_r(\lambda)$  tem um valor próprio  $\lambda$  de multiplicidade  $r$  e apenas um vector próprio  $x = e_1$  ( $e_1$  denota o vector  $(1, 0, \dots, 0)^T$  da base canónica).

**Teorema 1.7** Seja  $A$  uma matriz de ordem  $n$  com  $s$  valores próprios distintos  $\lambda_1, \lambda_2, \dots, \lambda_s$  de multiplicidades  $m_1, m_2, \dots, m_s$ , respectivamente. Então existe uma matriz não-singular  $R$  tal que  $R^{-1}AR$  tem blocos de Jordan  $J_r(\lambda_i)$  isolados ao longo da diagonal e todos os outros elementos iguais a zero. A soma das ordens das submatrizes associadas a cada valor próprio  $\lambda_i$  é  $m_i$ . A matriz  $R^{-1}AR$  é chamada **forma canónica de Jordan da matriz  $A$** , e é única a menos da ordem dos blocos ao longo da diagonal. O número total de blocos de Jordan na forma canónica de  $A$  é igual ao número de vectores próprios linearmente independentes de  $A$ .

**Dem.** Veja, por exemplo, [Hor85, pp 121 e segs].

**Exemplo 1.2** Uma matriz  $A$  de ordem 8 que possa ser reduzida à forma

$$\begin{aligned} J = R^{-1}AR &= \left( \begin{array}{ccc|cc|cc|c} \lambda_1 & 1 & & & & & & \\ & \lambda_1 & 1 & & & & & \\ & & \lambda_1 & & & & & \\ \hline & & & \lambda_1 & 1 & & & \\ & & & & \lambda_1 & & & \\ \hline & & & & & \lambda_2 & 1 & \\ & & & & & & \lambda_2 & \\ \hline & & & & & & & \lambda_3 \end{array} \right) \\ &= \begin{pmatrix} J_3(\lambda_1) & & & \\ & J_2(\lambda_1) & & \\ & & J_2(\lambda_2) & \\ & & & J_1(\lambda_3) \end{pmatrix} \end{aligned}$$

tem quatro vectores próprios independentes, dois correspondentes a  $\lambda_1$ , um correspondente a  $\lambda_2$  e outro correspondente a  $\lambda_3$ . Além disso, os vectores próprios de  $A$  são  $Re_1, Re_4, Re_6$  e  $Re_8$ . (Verifique!)

Os determinantes  $\det(J_r(\lambda_i - \lambda)) = (\lambda_i - \lambda)^r$  são chamados *divisores elementares* de  $A$ . Se  $r = 1$  o divisor elementar diz-se linear.

Uma matriz com valores próprios distintos tem todos os divisores elementares lineares. Se os valores próprios não são distintos, os divisores elementares podem ou não ser lineares, tendo-se, claramente, que  $A$  é deficiente se e só se pelo menos um dos seus divisores elementares é não-linear.

## 1.4 Propriedades de matrizes hermitianas

Seja  $A \in M_n$  uma matriz hermitiana. Então  $A$  satisfaz as seguintes propriedades:

**P1.** *Os valores próprios de  $A$  são reais.*

**Dem.** Com efeito, se  $Ax = \lambda x$  então  $\lambda = x^*Ax/x^*x$ . Mas, então,

$$\overline{\lambda} = \lambda^* = x^*A^*x/x^*x = \lambda,$$

uma vez que  $A^* = A$ .

**P2.**  *$A$  é não-deficiente e existe uma base de  $\mathbf{C}^n$  formada por vectores próprios ortonormados.*

**Dem.** Pelo Teorema de Schür, sabemos que existe uma matriz unitária  $P$  tal que  $P^*AP = U$ , com  $U$  triangular superior. Mas,

$$U^* = (P^*AP)^* = P^*A^*P = P^*AP = U,$$

o que implica que  $U = D$  com  $D$  uma matriz diagonal. Então,

$$P^*AP = D$$

e a matriz  $A$  é não-deficiente, pelo Teorema 1.5. Além disso, as colunas de  $P$  são os vectores próprios de  $A$  e, como  $P^*P = I$ , segue-se que  $x_i^*x_j = \delta_{ij}$ .

**P3.**  *$A$  é definida positiva sse os seus valores próprios são positivos.*

**Dem.** Existe uma matriz unitária  $P$  tal que  $P^*AP = \text{diag}(\lambda_i)$ . Então,

$$x^*Ax = x^*P\text{diag}(\lambda_i)P^*x = z^*\text{diag}(\lambda_i)z = \sum_{i=1}^n \lambda_i \bar{z}_i z_i,$$

onde  $z = P^*x$  e  $z_i$  designa a componente índice  $i$  do vector  $z$ . Assim,

$$x^*Ax > 0 \quad \forall x \neq 0 \iff \sum_{i=1}^n \lambda_i \bar{z}_i z_i > 0, \quad \forall z_i \iff \lambda_i > 0; \quad i = 1, \dots, n.$$

**Nota 1.3** Os resultados desta secção são, como é óbvio, válidos para matrizes reais simétricas.

## 1.5 Estimativas “a priori” para valores próprios

É conveniente, tanto do ponto de vista teórico, como do ponto de vista prático, determinar regiões limitadas do plano complexo contendo os valores próprios de uma dada matriz  $A$ . Tal informação fornece-nos aproximações iniciais para os valores próprios, as quais poderão depois ser melhoradas por utilização de métodos iterativos e desempenha, além disso, um papel importante no estudo de problemas de condicionamento (i.e., na análise das mudanças nos valores próprios de  $A$  induzidas por *pequenas* alterações nos seus elementos). Um resultado importante é-nos fornecido pelo seguinte teorema.

**Teorema 1.8 (Teorema dos círculos de Gerschgorin)** *Seja  $A = (a_{ij})$  uma matriz quadrada de ordem  $n$ . Consideremos os círculos  $C_i$ ;  $i = 1, \dots, n$ , de centros nos pontos  $a_{ii}$  e raios*

$$R_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|,$$

*isto é,*

$$C_i = \{\lambda \in \mathbb{C} : |\lambda - a_{ii}| \leq R_i\}.$$

*Então, os valores próprios de  $A$  estão na união dos círculos  $C_i$ . Estes círculos  $C_i$  são chamados círculos (ou discos) de Gerschgorin.*

**Dem.** Seja  $\lambda$  um valor próprio de  $A$  com  $x$  como vector próprio associado. Então,

$$Ax = \lambda x,$$

ou seja,

$$(\lambda - a_{ii})x_i = \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j; \quad i = 1, \dots, n.$$

Seja  $k$  tal que  $x_k$  é a componente de maior módulo de  $x$ , isto é, tal que  $|x_k| = \|x\|_\infty$ . Então,

$$|\lambda - a_{kk}| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \frac{|x_j|}{|x_k|} \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| = R_k,$$

isto é, o valor próprio  $\lambda$  está no círculo  $C_k$ . Podemos assim concluir que os valores próprios de  $A$  pertencem à união dos círculos  $C_i$ .  $\square$

**Corolário 1.1** *Se  $\lambda$  é valor próprio de  $A$ , então*

$$|\lambda - a_{jj}| \leq \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|,$$

*para pelo menos um  $j$ .*

**Corolário 1.2** *Uma matriz estritamente de diagonal dominante é não-singular.*

**Corolário 1.3** *Uma matriz hermitiana tal que*

$$a_{ii} > \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|; \quad i = 1, \dots, n,$$

*é definida positiva.*

(As demonstrações dos corolários anteriores ficam ao cuidado dos alunos.)

Pode provar-se ainda o seguinte teorema – para a sua demonstração, veja, por exemplo, [Wil65, p.71].

**Teorema 1.9** *Se a união de  $m$  círculos de Gerschgorin é disjunta dos restantes círculos, então essa união contém exactamente  $m$  valores próprios (contando multiplicidades). Em particular, se um círculo é disjunto dos restantes, ele contém exactamente um valor próprio.*

$\square$

**Melhoramento de Gerschgorin:** Suponhamos que o círculo  $C_k$  é disjunto dos restantes. Consideremos uma transformação de semelhança da forma  $DAD^{-1}$  onde

$$D = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & d & \\ & & & \ddots \\ & & & & 1 \end{pmatrix} \leftarrow \text{linha } k,$$

isto é,

$$DAD^{-1} = \begin{pmatrix} a_{11} & \dots & a_{1k}/d & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ da_{k1} & \dots & a_{kk} & \dots & da_{kn} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & a_{nk}/d & \dots & a_{nn} \end{pmatrix}$$

Os círculos de Gerschgorin são agora

$$|a_{ii} - \lambda| \leq \frac{1}{d}|a_{ik}| + \sum_{\substack{j \neq i \\ j \neq k}} |a_{ij}|, \quad i \neq k$$

e

$$|a_{kk} - \lambda| \leq d \left\{ \sum_{j \neq k} |a_{kj}| \right\}.$$

O objectivo do chamado *melhoramento de Gerschgorin* é determinar um valor de  $d$  que diminua o círculo  $C_k$ , mantendo-o disjunto dos restantes (os quais se tornam maiores à medida que  $d$  diminui).

**Definição 1.8** Dada uma matriz  $A$  e um vector  $x \neq 0$ , chama-se **quociente de Rayleigh de  $x$** , e denota-se por  $\mu_R$ , ao número

$$\mu_R := \frac{x^* Ax}{x^* x}.$$

**Nota 1.4** Se  $x = x_i$  é um vector próprio associado ao valor próprio  $\lambda_i$ , então

$$\mu_R = \frac{x_i^* Ax_i}{x_i^* x_i} = \frac{\lambda_i x_i^* x_i}{x_i^* x_i} = \lambda_i.$$

**Teorema 1.10** Se  $A$  é uma matriz hermitiana com valores próprios  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ , então, para qualquer vector  $x \neq 0$ , tem-se

$$\lambda_n \leq \mu_R \leq \lambda_1.$$

**Dem.** Seja  $x_i$ ;  $i = 1, \dots, n$ , uma base formada por vectores próprios ortonormados ( $A$  é hermitiana). Então, tem-se

$$x = \sum_{i=1}^n \alpha_i x_i,$$

donde

$$x^* Ax = \sum_{i=1}^n |\alpha_i|^2 \lambda_i$$

e

$$x^* x = \sum_{i=1}^n |\alpha_i|^2.$$

Logo,

$$\frac{x^*Ax}{x^*x} = \frac{\sum_{i=1}^n |\alpha_i|^2 \lambda_i}{\sum_{i=1}^n |\alpha_i|^2}.$$

Mas,

$$\lambda_n \sum_{i=1}^n |\alpha_i|^2 \leq \sum_{i=1}^n |\alpha_i|^2 \lambda_i \leq \lambda_1 \sum_{i=1}^n |\alpha_i|^2,$$

e o resultado segue-se de imediato.  $\square$

**Corolário 1.4**

$$\lambda_1 = \max_{x \neq 0} \frac{x^*Ax}{x^*x} \quad \text{e} \quad \lambda_n = \min_{x \neq 0} \frac{x^*Ax}{x^*x}.$$

$\square$

**Corolário 1.5** *Para qualquer matriz hermitiana  $A$  tem-se,*

$$\lambda_n \leq a_{ii} \leq \lambda_1; \quad i = 1, \dots, n.$$

**Dem.** Seja  $x = e_i$ . Então,

$$\frac{x^*Ax}{x^*x} = \frac{e_i^* A e_i}{e_i^* e_i} = a_{ii}.$$

$\square$

## 1.6 Breve referência a problemas de condicionamento

Consideremos agora a questão de saber quais os efeitos que pequenas perturbações nos elementos de uma matriz  $A$  podem produzir nos respectivos valores próprios; alguns valores próprios podem ser muito sensíveis a tais perturbações, sendo outros pouco sensíveis. Se pequenas alterações nos elementos de  $A$  podem induzir perturbações ilimitadas nalguns dos seus valores próprios, diremos que o problema da determinação desses valores próprios é *mal-condicionado*. Isto significa que, independentemente do método numérico utilizado, se torna muito difícil obter aproximações razoáveis para esses valores próprios, pois os inevitáveis erros de arredondamento introduzidos nos cálculos numéricos produzirão alterações significativas nos valores próprios calculados.

Seja  $E = (e_{ij})$  a matriz das perturbações introduzidas em  $A$  e suponhamos que

$$|e_{ij}| < \epsilon.$$

Estamos interessados em comparar os valores próprios  $\lambda_i$  de  $A$  com os valores próprios  $\mu_i = \mu_i(\epsilon)$  de  $A + E$ .



Consideremos primeiro o caso em que  $A$  é hermitiana e assumamos que  $E$  é também hermitiana. Então, pode provar-se o seguinte resultado – veja [Wil65, pp104 e segs]:

$$\left\{ \sum_{i=1}^n (\mu_i - \lambda_i)^2 \right\}^{1/2} \leq \|E\|_E \leq n\epsilon. \quad (1.10)$$

Conclui-se imediatamente de (1.10) que o problema da determinação dos valores próprios de uma matriz hermitiana é sempre bem condicionado.

Para matrizes não hermitianas, o problema é bem mais complexo. No entanto, os seguintes resultados são também demonstrados em [Wil65, pp 67 e segs].

Se  $\lambda_1$  é um valor próprio *simples* de  $A$ , então

$$\mu_1(\epsilon) - \lambda_1 = k_1\epsilon + k_2\epsilon^2 + \dots = k_1\epsilon + \mathcal{O}(\epsilon^2), \quad (1.11)$$

onde  $k_1, k_2, \dots$ , designam constantes independentes de  $\epsilon$ . Assim, para  $\epsilon$  suficientemente pequeno, a principal perturbação em  $\lambda_1$  é  $k_1\epsilon$ . Além disso, se  $A$  tem apenas divisores elementares lineares e se  $x_1$  e  $y_1$  são respectivamente um vector próprio à direita e um vector próprio à esquerda de  $A$  associados a  $\lambda_1$  e normalizados de tal forma que

$$\|x_1\|_2 = \|y_1\|_2 = 1,$$

então pode estabelecer-se o seguinte limite superior para  $k_1$ :

$$|k_1| \leq \frac{n}{|y_1^T x_1|}. \quad (1.12)$$

**Nota 1.5** •  $|y_1^T x_1| \leq \|y_1\|_2 \|x_1\|_2 = 1$ .

- Se  $x_1$  e  $y_1$  são reais, então  $y_1^T x_1$  é o co-seno do ângulo  $\theta$  entre os vectores  $y_1$  e  $x_1$ .
- O valor de  $\theta$  acima referido pode ser arbitrariamente pequeno.
- Se  $A$  é hermitiana, então  $y_1^T x_1 = 1$ , o que confirma a afirmação de que o problema do cálculo dos seus valores próprios é bem condicionado.

Para um estudo bastante pormenorizado sobre o condicionamento, nos restantes casos, veja [Wil65, Capítulo 2]. Referimos apenas aqui o seguinte:

- O problema da determinação dos valores próprios correspondentes a divisores elementares não-lineares é, em geral, mal-condicionado.

- A análise da sensibilidade dos vectores próprios de uma matriz relativamente a perturbações nos seus elementos é bastante complexa. Pode provar-se que, mesmo quando  $A$  é hermitiana, o problema da determinação de vectores próprios correspondentes a valores próprios "próximos" é mal-condicionado.

## 1.7 Notas e referências adicionais

- Para uma cobertura mais completa sobre a teoria básica de valores e vectores próprios, veja, por exemplo, [Gan54]
- Taussky [Tau49] dá-nos uma discussão histórica do Teorema de Gerschgorin. A técnica do melhoramento de Gerschgorin apareceu no artigo original deste autor [Ger31] e foi posteriormente estudada por Varga [Var65].
- Para uma extensão do Teorema 1.10, veja, por exemplo, [Hou64].
- O problema do comportamento dos valores e vectores próprios de uma matriz face a uma perturbação nos seus elementos foi objecto de muitos estudos. Para um tratamento pormenorizado deste assunto, veja, por exemplo [Wil63], [Wil65] e [Hou64]. Em particular, em [Wil65] é feito um estudo do problema baseado na forma canónica de Jordan e no Teorema dos círculos de Gerschgorin.

## 1.8 Exercícios

1. Mostre que o polinómio característico  $p(\lambda) = \det(A - \lambda I)$  de uma matriz  $A = (a_{ij})$  tem a forma

$$p(\lambda) = (-1)^n \lambda^n + (-1)^{n-1} (a_{11} + a_{22} + \cdots + a_{nn}) \lambda^{n-1} + \sum_{k=n-2}^0 A_k \lambda^k.$$

Conclua que  $\sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i$ . ( $\sum_{i=1}^n a_{ii}$  diz-se o traço de  $A$ )

2.
  - a) Mostre, através de exemplos, que se  $\lambda_i$  e  $\mu_i$  são respectivamente os valores próprios de duas matrizes  $A$  e  $B$  da mesma ordem, então os valores próprios de  $AB$  e  $A + B$  não são necessariamente da forma  $\lambda_i \mu_i$  e  $\lambda_i + \mu_i$ .
  - b) Mostre que, se  $x$  é vector próprio de  $A$  associado ao valor próprio  $\lambda$  e também é vector próprio de  $B$  associado ao valor próprio  $\mu$ , então  $\lambda \mu$  é valor próprio de  $AB$  e  $\lambda + \mu$  é valor próprio de  $A + B$ .

3. a) Prove que as matrizes

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

têm o mesmo polinómio característico, mas não são semelhantes.

- b) Diga, justificando, se as seguintes matrizes são semelhantes.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

- c) Mostre que matrizes semelhantes têm o mesmo polinómio característico.

4. Considere o polinómio mónico

$$p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0.$$

- a) Mostre que

$$p(\lambda) = (-1)^n \det(A - \lambda I),$$

onde  $A$  é a matriz

$$A = \begin{bmatrix} 0 & & & & -a_0 \\ 1 & 0 & & & -a_1 \\ & 1 & 0 & & -a_2 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 0 & -a_{n-2} \\ & & & & 1 & -a_{n-1} \end{bmatrix}$$

- b) Conclua que os zeros do polinómio  $p$  são os valores próprios da matriz  $A$ .

A matriz  $A$  definida anteriormente é chamada *matriz companheira do polinómio  $p$* .



5. Considere os polinómios,

$$p_1(x) = x^3 - 4x^2 + 2x - 1 \quad \text{e} \quad p_2(x) = x^5 + 4x^4 - 2x^2 + x - 3.$$

- a) Obtenha a matriz companheira de  $p_1$  e  $p_2$ , usando a definição anterior.
- b) Use a função *compan* do MATLAB para resolver a alínea anterior. Compare as matrizes obtidas.
- c) Confirme, usando a função *poly* do MATLAB, que  $p_1$  e  $p_2$  são, de facto, os polinómios característicos das suas matrizes companheiras. Alternativamente, pode usar as funções *roots* e *eig* para verificar que os zeros de cada um dos polinómios dados são os valores próprios da sua matriz companheira.

(Refira-se, a propósito, que o método implementado pela função *roots* do MATLAB para calcular os zeros de um dado polinómio passa exactamente pelo cálculo dos valores próprios da sua matriz companheira.)

6. Seja  $P$  uma matriz unitária. Prove que, se  $\lambda$  é um valor próprio de  $P$ , então  $1/\bar{\lambda}$  é também um valor próprio de  $P$  e, além disso,  $|\lambda| = 1$ .
7. Prove que, se  $A$  tem  $n$  valores próprios distintos e  $X$  e  $Y$  são matrizes cujas colunas são, respectivamente, vectores próprios à direita e à esquerda de  $A$ , então

$$Y^T X = I,$$

desde que se multipliquem os vectores próprios por escalares convenientes.

8. Seja  $A$  uma matriz não deficiente com valores próprios  $\lambda_1, \dots, \lambda_n$  e  $X$  e  $Y$  matrizes cujas colunas  $x_i$  e  $y_i, i = 1, \dots, n$  são, respectivamente, vectores próprios à direita e à esquerda de  $A$ . Mostre que,

$$A = \sum_{i=1}^n \lambda_i x_i y_i^T.$$

Esta decomposição designa-se por *decomposição espectral* da matriz  $A$ .



9. Obtenha a decomposição espectral da matriz

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -1 & 1 & 3 \\ 2 & 3 & 2 \end{bmatrix}.$$

10. Uma matriz  $A \in M_n$  diz-se *normal* sse  $A^* A = A A^*$ .
- a) Mostre que as únicas matrizes triangulares que são normais são as matrizes diagonais.
  - b) Mostre que toda a matriz triangular  $T$  unitariamente semelhante a uma matriz  $A$  é normal se e só se  $A$  é normal.
  - c) Mostre que as seguintes afirmações são equivalentes.
    - (i)  $A$  é normal.
    - (ii)  $A$  é unitariamente semelhante a uma matriz diagonal.
    - (iii) Existe uma base de  $\mathbb{C}^n$  formada por vectores próprios de  $A$  ortonormados.

11. Seja  $A$  uma matriz tridiagonal de ordem  $n$  tal que

$$a_{ii} = 0; \quad i = 1, \dots, n, \quad a_{i,i+1} = a_{i+1,i} = 1, \quad i = 1, \dots, n-1.$$

Para  $j = 1, \dots, n$ , seja  $x^{(j)}$  o vector cuja  $i$ -ésima componente é:

$$x_i^{(j)} = \sin [ij\pi/(n+1)]; \quad i = 1, \dots, n.$$

Mostre que

$$Ax^{(j)} = 2 \cos[j\pi/(n+1)]x^{(j)}; \quad j = 1, \dots, n.$$

**12.** Mostre que, se  $A$  é uma matriz tridiagonal tal que

$$a_{ii} = d; \ i = 1, \dots, n, \quad a_{i,i+1} = a_{i+1,i} = e, \ i = 1, \dots, n-1,$$

então os valores próprios de  $A$  são

$$\lambda_j = d + 2e \cos[j\pi/(n+1)]; \ j = 1, \dots, n.$$

**13.** Mostre que

$$\|A\|_2 = \{\rho(A^*A)\}^{1/2}.$$

**14.** Seja  $P \in M_n$  uma matriz *unitária*. Mostre que:

a)  $\|Px\|_2 = \|x\|_2, \ \forall x \in \mathbb{C}^n.$

b)  $\|PA\|_2 = \|A\|_2, \ \forall A \in M_n.$

c)  $\|P^*AP\|_2 = \|A\|_2, \ \forall A \in M_n.$

d) Se  $A$  é uma matriz hermitiana, então  $P^*AP$  também é hermitiana.

**15.** Seja  $A \in M_n$ .

a) Mostre que, para qualquer norma matricial compatível com uma norma vectorial,

$$\rho(A) \leq \|A\|.$$

b) Seja

$$R_i = \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}|.$$

Mostre que,

$$\rho(A) \leq \max_i \{|a_{ii}| + R_i\},$$

e, se  $A$  é não singular,

$$\{\rho(A^{-1})\}^{-1} \geq \min_i \{|a_{ii}| - R_i\}.$$

c) Prove que, se  $A$  é não singular e se  $\lambda_1, \dots, \lambda_n$  são os valores próprios de  $A$ , então,

$$\{\rho((A^*A)^{-1})\}^{-1} \leq |\lambda_i|^2 \leq \rho(A^*A); \ i = 1, \dots, n.$$

**16.** Obtenha a seguinte estimativa

$$|\lambda - 1.0| < 10^{-4},$$

para o menor valor próprio da matriz

$$A = \begin{bmatrix} 6 & 1 & 0 & 0 \\ 1 & 5 & 1 & 0 \\ 0 & 1 & 4 & 10^{-3} \\ 0 & 0 & 10^{-3} & 1 \end{bmatrix}.$$

**17.** Considere a matriz

$$A = \begin{bmatrix} 1 & 0.1 & 0 \\ 0.1 & 10 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

- a) Indique intervalos reais que contenham os valores próprios de  $A$ . (Note que, sendo  $A$  real e simétrica, os seus valores próprios são reais)
- b) Isole cada um dos valores próprios de  $A$  num intervalo de amplitude o mais pequena possível.

**18.** Considere a seguinte matriz tridiagonal de ordem 5:

$$A = \begin{bmatrix} \alpha + \beta & \beta & & & \\ & \beta & \alpha & \beta & \\ & & \beta & \alpha & \beta \\ & & & \beta & \alpha & \beta + \varepsilon \\ & & & & \beta + \varepsilon & \alpha + \beta \end{bmatrix},$$

onde  $\alpha, \beta, \varepsilon$  são números positivos e sejam  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ , os valores próprios de  $A$ .  
Mostre que:

$$\alpha + 2\beta + 0.4\varepsilon \leq \lambda_1 \leq \alpha + 2\beta + \varepsilon.$$

(Sugestão: Use o Teorema 1.8 e o Teorema 1.10)

**19.** Seja  $A$  a matriz

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 3 & 2 & 0 \\ 0 & 2 & 5 & \varepsilon \\ 0 & 0 & \varepsilon & 9 \end{bmatrix},$$

com  $-1 \leq \varepsilon \leq 1$  e seja  $\rho(A)$  o raio espectral de  $A$ . Mostre que

$$9 \leq \rho(A) \leq 9 + \frac{\varepsilon^2}{2}.$$

**20.** Justifique que o maior valor próprio  $\lambda$  da matriz

$$A = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1.5 & 0 \\ 1 & 2 & 5 \end{bmatrix},$$

satisfaz  $|\lambda - 5| < 0.8$ .

21. Escreva uma função em MATLAB *gersch.m* para obter os discos de Gerschgorin de uma dada matriz.
22. Escreva uma função em MATLAB *pgersch.m* para representar geometricamente a região do plano complexo que contém os valores próprios de uma dada matriz  $A$ .
23. Use a função *pgersch* para localizar geometricamente os valores próprios das seguintes matrizes.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & -1 & 1 \\ 3 & 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -1 & 0 \\ 2 & 1 & 1 \\ -1 & -1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} i & 1 & 1+i \\ -1 & 2i & 2-i \\ -i & i & 1 \end{bmatrix}.$$

24. Considere as matrizes

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 10^{-3} & 0 & 0 & 0 & 1 \end{bmatrix}$$

- a) Use a função *eig* do MATLAB para obter os valores próprios da matriz  $\tilde{A}$ .
- b) Compare os valores próprios das matrizes  $A$  e  $\tilde{A}$  e comente.

25. Considere as matrizes

$$A = \begin{bmatrix} 1 & 0 & 0 & 10^{-3} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \quad \text{e} \quad \tilde{A} = \begin{bmatrix} 1 & 0 & 0 & 10^{-3} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 10^{-3} & 0 & 0 & 3 \end{bmatrix}$$

- a) Verifique que 1 e 3 são valores próprios de  $A$  e que o seu polinómio característico é:

$$p(\lambda) = \lambda^4 - 6\lambda^3 + 12\lambda^2 - 10\lambda + 3.$$

(Use a função *poly* do MATLAB)

- b) Calcule os valores próprios de  $\tilde{A}$ , usando a função *eig* do MATLAB.
- c) Calcule os zeros do polinómio

$$p^*(\lambda) = (1 + 10^{-3})\lambda^4 - 6\lambda^3 + 12\lambda^2 - 10\lambda + 3.$$

(Use a função *roots* do MATLAB)

- d) Comente os resultados obtidos nas alíneas anteriores.

26. Seja  $I$  a matriz identidade de ordem 2 e seja  $E$  a seguinte matriz das perturbações introduzidas em  $I$ :

$$E = \begin{bmatrix} \epsilon & \delta \\ 0 & 0 \end{bmatrix}.$$

- a) Escreva uma função em MATLAB *perturb.m* com parâmetros de entrada *epsilon* e *delta* e cujos parâmetros de saída *plambda* e *px* contêm, respectivamente, os valores e vectores próprios da matriz  $I + E$ .
- b) Compare os valores e vectores próprios da matriz  $I$  e da matriz perturbada  $I + E$ , para vários valores (pequenos) dos parâmetros *epsilon* e *delta*.
- c) Comente a sensibilidade dos valores e vectores próprios da matriz  $I$ , face a pequenas perturbações dos seus coeficientes.



## 2. Matrizes de Transformação

À primeira vista, poder-se-á pensar que o problema do cálculo dos valores próprios de uma matriz será facilmente resolvido, calculando os coeficientes do polinómio característico e determinando, em seguida, os seus zeros, por qualquer dos processos já estudados para a determinação de raízes de equações polinomiais – veja, por exemplo [Val90, pp 51 e segs]. Contudo, devemos notar o seguinte:

- O cálculo dos coeficientes do polinómio característico através da definição de determinante envolve, como se sabe, um muito elevado número de operações, não sendo por isso de recomendar. Outros métodos mais eficientes para o cálculo dos coeficientes do polinómio característico podem ser vistos, por exemplo, em [RR78, pp 485,486] e [Hou64, pp 149 e segs].
- Como sabemos, erros relativamente pequenos nos coeficientes de um polinómio podem induzir grandes erros nas suas raízes, sobretudo se o polinómio tiver grau elevado. Assim, ainda que os coeficientes do polinómio característico tenham sido calculados adequadamente, corremos o risco de transformar um problema inicialmente bem condicionado – o do cálculo dos valores próprios de uma certa matriz – num problema mal condicionado – o do cálculo dos zeros de um determinado polinómio.

Em conclusão, o cálculo dos valores próprios de uma matriz, directamente através da definição de polinómio característico, é de evitar, a não ser para matrizes de reduzida dimensão.

A propriedade de invariância dos valores próprios por transformações de semelhança que referimos no capítulo anterior (Teorema 1.4) sugere a possibilidade de determinar os valores próprios de  $A$ , calculando os de uma matriz semelhante  $SAS^{-1}$  que possua uma forma mais adequada para esse fim. Esta é, basicamente, a ideia de diversos algoritmos que iremos descrever em capítulos seguintes. As suas diferenças estão, essencialmente, na estrutura da matriz a que se pretende reduzir  $A$  por semelhança. Na prática, em vez de calcular uma única matriz de transformação

$S$ , procede-se de uma forma *iterativa*, reduzindo-se gradualmente a matriz original a uma forma mais simples. Neste capítulo, estudaremos a redução de uma matriz hermitiana à forma tridiagonal, usando transformações de semelhança. No caso em que a matriz original é não hermitana, a matriz resultante terá a forma de Hessenberg. As matrizes básicas utilizadas nestas reduções são vulgarmente chamadas *matrizes elementares*.

## 2.1 Matrizes elementares

**Definição 2.1 (Matriz elementar de permutação  $I_{ij}$ )** Denotaremos por  $I_{ij}$  a matriz resultante da matriz identidade por troca das suas linhas  $i$  e  $j$ . Uma matriz deste tipo chama-se **matriz elementar de permutação**.

**Definição 2.2** Chama-se **matriz de permutação** a toda a matriz que possa ser obtida como produto de matrizes elementares de permutação.

**Definição 2.3 (Matriz elementar não-unitária  $M_k$ - eliminação Gaussiana)** Denota-se por  $M_k$  uma matriz obtida da matriz identidade substituindo os elementos na coluna  $k$ , abaixo da posição  $(k, k)$ , por elementos  $-m_{ik}$ , não todos nulos. Uma matriz deste tipo chama-se **matriz elementar não-unitária** ou **matriz de Gauss**.

**Exemplo 2.1** No conjunto das matrizes  $3 \times 3$ ,

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} \quad M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}$$

### Propriedades das matrizes de Gauss

1.  $M_k = I - m^{(k)} e_k^T$ , onde  $m^{(k)} = (0 \cdots 0 \ m_{k+1,k} \cdots m_{n,k})^T$  e  $e_k \equiv$  coluna  $k$  da matriz identidade.
2. A pré-multiplicação de  $A$  por  $M_k$  corresponde a multiplicar a linha  $k$  por  $m_{ik}$  e a subtraí-la da linha  $i$ ;  $i = k + 1, \dots, n$ .
3.  $M_k^{-1} = I + m^{(k)} e_k^T$ .

**Definição 2.4 (Matriz (real) de rotação plana - Givens)** Denota-se por  $R_{ij}(\theta)$ , ou mais usualmente, apenas por  $R_{ij}$ , uma matriz que se obtém da matriz identidade substituindo os elementos nas posições  $(i, i)$ ,  $(i, j)$ ,  $(j, i)$ ,  $(j, j)$  respectivamente por elementos  $r_{ii}$ ,  $r_{ij}$ ,  $r_{ji}$  e  $r_{jj}$  tais que

$$r_{ii} = r_{jj} = \cos \theta, \quad r_{ij} = -r_{ji} = \sin \theta,$$

para um certo valor  $\theta \in \mathbb{R}$ . Uma matriz deste tipo chama-se **matriz (real) de rotação plana** ou **matriz de Givens**.

Temos então

$$R_{ij} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos \theta & \dots & \sin \theta & \\ & & \vdots & & \vdots & \\ & & -\sin \theta & \dots & \cos \theta & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix}$$

**Nota 2.1** Por vezes, usaremos uma notação condensada

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

para designar a matriz  $R_{ij}$  acima referida.

### Propriedades das matrizes de Givens

1.  $R_{ij}$  é uma matriz ortogonal, i.e.

$$R_{ij}^T R_{ij} = I.$$

2. A matriz  $\begin{pmatrix} c & s \\ -s & c \end{pmatrix}$  corresponde a uma rotação do plano  $ij$  de um ângulo  $\theta$ .

(Em que sentido?)

**Definição 2.5 (Matriz hermitiana elementar P - Householder)** A uma matriz do tipo

$$P = I - 2ww^*$$

onde  $w \in \mathbb{C}^n$  é tal que  $w^*w = 1$ , chamamos **matriz hermitiana elementar** ou **matriz de Householder**.

## Propriedades das matrizes de Householder

1.  $P$  é hermitiana.

Dem.

$$P^* = (I - 2ww^*)^* = I - 2(w^*)^*w^* = I - 2ww^* = P,$$

□

2.  $P$  é unitária.

Dem.

$$\begin{aligned} P^*P &= (I - 2ww^*)(I - 2ww^*) \\ &= I - 4ww^* + 4ww^*ww^* \\ &= I, \end{aligned}$$

□

3. Em geral, na prática,  $w \in \mathbb{R}^n$ ,  $P = I - 2ww^T$  com  $w^T w = 1$ . Nesse caso,  $P$  é simétrica e ortogonal.

4. Se  $w = (0 \cdots 0 \ w_{k+1} \cdots w_n)^T = (0 \mid w')^T$ , com  $w' = (w_{k+1} \cdots w_n)^T$ , então a matriz

$$P = P_k = I - 2ww^*,$$

pode escrever-se na forma particionada seguinte:

$$P_k = \left( \begin{array}{c|c} I_k & O \\ \hline O & I_{n-k} - 2w'w'^* \end{array} \right).$$

## 2.2 Utilização de matrizes elementares na redução de um vector coluna

Suponhamos que se pretende reduzir um dado vector coluna  $a = (a_1, a_2, \dots, a_n)^T$  à forma  $(k, 0, \dots, 0)^T = ke_1$ . Vejamos como tal objectivo pode ser atingido, pré-multiplicando  $a$  por matrizes elementares.

### (i) Eliminação Gaussiana

Seja

$$M_1 = \left( \begin{array}{cccc} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & \dots & 1 \end{array} \right)$$

uma matriz elementar não-unitária com

$$m_{i1} = a_i/a_1; \quad i = 2, \dots, n.$$

Então,

$$M_1 a = \begin{pmatrix} a_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

**Nota 2.2** Se  $a_1 = 0$ , o método falha. Também, se  $|a_1|$  é *pequeno* relativamente às restantes componentes de  $a$ , o método é instável. Para evitar isso, usa-se *técnica de escolha de pivot*. Mais precisamente, troca-se a linha  $l$  com a linha 1, sendo  $a_l$  a componente de maior módulo de  $a$ . Nesse caso temos:

**Método estabilizado:**

$$\begin{aligned} I_{1l} a &= a' \\ M_1 a' &= k e_1 \quad \text{onde } k = a'_1. \end{aligned}$$

Para eliminar elementos abaixo de  $a_k$  deixando os restantes elementos inalterados, podemos usar  $a_k$  para a eliminação, i.e. efectuar o produto  $M_k a$  (ou  $M_k I_{kl} a$ , no método estabilizado).

$$M_k a = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{k+1,k} & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & -m_{n,k} & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

O vector  $m^{(k)} = (0 \cdots 0 \ m_{k+1,k} \cdots m_{n,k})^T$  é chamado o *vector de Gauss* e os elementos de  $m^{(k)}(k+1 : n)$  são chamados os *multiplicadores*.

Apresentamos de seguida um algoritmo para redução de um vector  $a$  à forma  $k e_1$ , usando matrizes de Gauss. Neste algoritmo, tal como em todos os outros destes apontamentos, é usada uma notação apropriada para uma fácil implementação em MATLAB.

**Algoritmo 2.1 Matriz de Gauss  $M_1$** 

```

% Este algoritmo constrói o vector dos multiplicadores
% necessários à redução de um vector  $a \in \mathbb{R}^n$  à forma  $ke_1$ 
% Parâmetros de entrada: vector  $a$ 
% Parâmetros de saída: vector  $mul$  dos multiplicadores

```

```

função  $mul = \text{gauss}(a)$ 
     $mul = a(2 : n) / a(1);$ 
fim função gauss

```

```

% Este algoritmo envolve  $n - 1$  flops

```

No algoritmo anterior, a matriz  $M_1$  não é explicitamente formada para economizar espaço. Relembremos que

$$M_1 = I - m^{(1)} e_1^T, \text{ onde } m^{(1)} = [0 \mid mul^T]^T,$$

pelo que  $M_1$  pode ser construída, se necessário, muito facilmente.

No caso de se pretender calcular a matriz  $M_k$ , então  $mul = \text{gauss}(a(k : n))$  é o vector que contém os multiplicadores e a matriz  $M_k$  pode ser obtida de  $M_k = I - m^{(k)} e_k^T$ , onde  $m^{(k)} = [0 \cdots 0 \mid mul^T]^T$ .

A pré-multiplicação de uma matriz  $A$  por uma matriz de Gauss pode ser obtida muito facilmente. De facto,

$$M_k A = (I - m^{(k)} e_k^T) A = A - m^{(k)} (e_k^T A).$$

O vector  $e_k^T A$  não é mais que a linha  $k$  da matriz  $A$ . Além disso, como  $m^{(k)}(0 : k) = 0$ , ao efectuar o produto  $M_k A$ , só  $A(k + 1 : n, :)$  é afectada, i.e. as linhas 1 a  $k$  de  $A$  permanecem inalteradas. Temos assim, o seguinte algoritmo.

**Algoritmo 2.2 Pré-multiplicação de  $A \in \mathbb{R}^{n \times r}$  por  $M_1$** 

```

% Parâmetros de entrada: matriz  $A$ , vector dos multiplicadores  $mul$ 
% Parâmetros de saída: matriz  $A$  ( $A = M_1 * A$ )

```

```

função  $A = \text{pre\_gauss}(A, mul)$ 
     $A(2 : n, :) = A(2 : n, :) - mul * A(1, :);$ 
fim função pre_gauss

```

```

% Este algoritmo envolve  $2(n - 1)r$  flops

```

**(ii) Transformação de rotação plana (Givens)**

Suponhamos que  $a \in \mathbb{R}^n$ . Consideremos

$$R_{1j}a = \begin{pmatrix} \cos \theta & \dots & \sin \theta & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ -\sin \theta & \dots & \cos \theta & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_j \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \cos \theta + a_j \sin \theta \\ \vdots \\ -a_1 \sin \theta + a_j \cos \theta \\ \vdots \\ a_n \end{pmatrix}.$$

Pretendemos determinar  $\theta$  de tal modo que

$$-a_1 \sin \theta + a_j \cos \theta = 0,$$

isto é, tal que

$$\frac{\sin \theta}{\cos \theta} = \frac{a_j}{a_1}. \quad (2.1)$$

Segue-se imediatamente de (2.1) que podemos tomar

$$\sin \theta = \frac{a_j}{\{a_1^2 + a_j^2\}^{1/2}} \quad \text{e} \quad \cos \theta = \frac{a_1}{\{a_1^2 + a_j^2\}^{1/2}}.$$

**Nota 2.3** • Se  $a_1^2 + a_j^2 = 0$ , não será necessária qualquer transformação.

- O método consiste na utilização de  $(n - 1)$  destas matrizes  $R_{1j}$ ;  $j = 2, \dots, n$ .
- Tem-se  $k = a_1 \cos \theta + a_j \sin \theta = \sqrt{a_1^2 + a_j^2} > 0$ .
- Se  $a \in \mathbb{C}^n$ , a redução à forma  $(k, 0, \dots, 0)^T$  com  $k \in \mathbb{R}, k \geq 0$ , pode fazer-se, utilizando para isso matrizes de rotação plana *complexas*, da forma

$$\begin{pmatrix} e^{i\alpha} \cos \theta & \dots & e^{i\beta} \sin \theta & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ -e^{-i\beta} \sin \theta & \dots & e^{-i\alpha} \cos \theta & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \dots 1 \end{pmatrix}.$$

Estas matrizes são matrizes unitárias.

**Algoritmo 2.3 Matriz de Givens  $R_{1,2}$** 

```
% Cálculo da matriz de Givens  $R_{1,2}(\theta)$ 
% para redução de um vector  $a = (a_1 \ a_2)^T$  à forma  $ke_1$ 
% Parâmetros de entrada: componentes  $a_1$  e  $a_2$  do vector  $a$ 
% Parâmetros de saída:  $c = \cos \theta$ ,  $s = \sin \theta$ 
```

```
função [c, s]=givens(a1, a2)
    aux =  $\sqrt{a1^2 + a2^2}$ ;
    c = a1/aux;
    s = a2/aux;
fim função givens
```

```
% Este algoritmo envolve 5 flops
```

**Nota 2.4** Em [GVL97], é apresentado um algoritmo alternativo para calcular  $c$  e  $s$ . Esse algoritmo tem especial atenção a possíveis problemas de *overflow*.

Relembremos que as matrizes elementares são usadas para reduzir uma dada matriz  $A$ , através de transformações de semelhança, a uma forma mais adequada para o cálculo dos seus valores próprios. Como as matrizes de Givens são ortogonais ( $R_{ij}^{-1} = R_{ij}^T$ ), então a matriz  $R_{ij}AR_{ij}^T$  é semelhante à matriz  $A$ .

A estrutura simples das matrizes de Givens  $R_{ij}$  deve ser explorada quando se pretende fazer os produtos  $R_{ij}A$  ou  $AR_{ij}^T$ . De facto, apenas as linhas  $i$  e  $j$  de  $A$  são afectadas pela pré-multiplicação de  $A$  por  $R_{ij}$  e as colunas  $i$  e  $j$  pela pós-multiplicação de  $A$  por  $R_{ij}^T$ .

**Algoritmo 2.4 Pré-multiplicação de  $A \in \mathbb{R}^{2 \times n}$  por  $R_{12}$** 

```
% Parâmetros de entrada: matriz  $A$ ,  $c$ ,  $s$ 
% Parâmetros de saída: matriz  $A$  ( $A = R_{12} * A$ )
```

```
função A=pre_givens(A, c, s)
    for j = 1 : n
        x1 = A(1, j); x2 = A(2, j);
        A(1, j) = c * x1 + s * x2;
        A(2, j) = -s * x1 + c * x2;
    end
fim função pre_givens
```

```
% Este algoritmo envolve  $6n$  flops
```



**Algoritmo 2.5 Pós-multiplicação de  $A \in \mathbb{R}^{n \times 2}$  por  $R_{12}^T$** 

```

% Parâmetros de entrada: matriz A, c, s
% Parâmetros de saída: matriz A ( $A = A * R_{12}^T$ )

função A=pos_givens(A, c, s)
    for i = 1 : n
        x1 = A(i, 1); x2 = A(i, 2);
        A(i, 1) = c * x1 + s * x2;
        A(i, 2) = -s * x1 + c * x2;
    end
fim função pos_givens

% Este algoritmo envolve 6n flops

```

Denotando por  $A([i \ j], :)$  as linhas  $i$  e  $j$  da matriz  $A$ , facilmente se conclui que

$$A([i \ j], :) = \mathbf{pre\_givens}(A([i \ j], :), c, s),$$

faz a atribuição  $A := R_{ij}A$ . (O que faz  $A(:, [i \ k]) = \mathbf{pos\_givens}(A(:, [i \ k]), c, s)$ ?)

**(iii) Transformação hermitiana elementar (Householder)**

Suponhamos novamente que  $a \in \mathbb{R}^n$ . Pretende determinar-se  $w \in \mathbb{R}^n$  tal que  $w^T w = 1$  e de tal modo que

$$Pa = (I - 2ww^T)a = ke_1. \quad (2.2)$$

Se tal for possível, teremos

$$k^2 = k^2 e_1^T e_1 = (Pa)^T (Pa) = a^T P^T Pa = a^T a,$$

uma vez que  $P^T P = I$ . Assim, teremos,

$$k = \pm \{a^T a\}^{1/2} = \pm \left\{ \sum_{i=1}^n a_i^2 \right\}^{1/2}. \quad (2.3)$$

(Note que há duas escolhas possíveis para  $k$ .) Segue-se imediatamente de (2.2) que

$$2ww^T a = a - ke_1. \quad (2.4)$$

Sejam

$$u := a - ke_1 \quad \text{e} \quad \alpha := 2w^T a. \quad (2.5)$$

Temos então,

$$\alpha w = a - k e_1 = u. \quad (2.6)$$

(O sinal de  $k$  é usualmente escolhido de forma a evitar cancelamento subtrativo no cálculo da primeira componente  $a_1 - k$  de  $u$ .)

Como  $w^T w = 1$ , tem-se, em virtude de (2.6),

$$\alpha^2 = \alpha^2 w^T w = u^T u. \quad (2.7)$$

Conclusão:

$$P = I - 2ww^T = I - \frac{2}{\alpha^2}uu^T.$$

**Nota 2.5** Se  $a \in \mathbb{C}^n$ , para reduzir  $a$  à forma  $ke_1$  deverão ser utilizadas matrizes da forma  $I - 2ww^*$  com  $w \in \mathbb{C}^n$ . Nestas condições,  $k \in \mathbb{C}$ . Se pretendermos reduzir a coluna  $a \in \mathbb{C}^n$  à forma  $ke_1$  com  $k$  real, deverão ser utilizadas matrizes mais gerais da forma  $I - \mu ww^*$  com  $\mu \in \mathbb{C}$ ; [Wil65, p.49,50]. Com  $\mu \neq 2$ ,  $P$ , se for unitária, já não é hermitiana – veja Exercício 3.

#### Algoritmo 2.6 Matriz de Householder $P$

```
% Cálculo da matriz de Householder  $P = I - 2ww^T$ 
% para redução de um vector  $a \in \mathbb{R}^n$  à forma  $ke_1$ 
% Parâmetros de entrada: vector  $a$ 
% Parâmetros de saída: vector  $w$ 

função  $w = \text{house}(a)$ 
     $k = \|a\|_2$ ;
     $u = a$ ;
     $u(1) = a(1) + \text{sign}(a(1)) * k$ ;
     $alfa = \|u\|_2$ ;
     $w = u / alfa$ ;
fim função house

% Este algoritmo envolve  $5n$  flops
```

**Nota 2.6** Na prática, pode calcular-se apenas o vector  $u$  para poupar o trabalho de calcular  $w$  e usar a fórmula  $P = I - 2 \frac{uu^T}{u^T u}$  em vez de  $P = I - 2ww^T$ . Esta última fórmula requer o cálculo de  $\alpha$  (e, consequentemente, de uma raiz quadrada), enquanto a primeira só necessita do valor de  $\alpha^2$ .

A matriz  $PAP$  é semelhante a  $A$ , uma vez que  $P$  é uma matriz simétrica e ortogonal. A estrutura simples das matrizes de Householder deve também ser explorada quando se pretende efectuar os produtos  $PA$  ou  $AP$ . Como

$$PA = (I - 2ww^T)A = A - 2w(A^T w)^T$$

e

$$AP = A(I - 2ww^T) = A - 2(Aw)w^T,$$

definindo

$$v = -2A^T w \quad \text{e} \quad u = -2Aw,$$

os produtos  $PA$  e  $AP$  podem ser escritos como

$$PA = A + vw^T \quad \text{e} \quad AP = A + uw^T,$$

não sendo, por isso, necessário determinar explicitamente a matriz  $P$ .

Os algoritmos seguintes implementam a pré e pós multiplicação de uma matriz  $A \in \mathbb{R}^{m \times n}$  por uma matriz de Householder  $P$ , uma vez conhecido o vector  $w$  tal que  $P = I - 2ww^T$ .

**Algoritmo 2.7 Pré-multiplicação de  $A \in \mathbb{R}^{m \times n}$  por  $P$**

% **Parâmetros de entrada:** matriz  $A$ , vector  $w$   
 % **Parâmetros de saída:** matriz  $A$  ( $A = P * A$ )

```
função A=pre_house(A, w)
    v = -2A^T w;
    A = A + vw^T;
fim função pre_house
```

% Este algoritmo envolve  $4mn$  flops

**Algoritmo 2.8 Pós-multiplicação de  $A \in \mathbb{R}^{m \times n}$  por  $P$**

% **Parâmetros de entrada:** matriz  $A$ , vector  $w$   
 % **Parâmetros de saída:** matriz  $A$  ( $A = A * P$ )

```
função A=pos_house(A, w)
    u = -2Aw;
    A = A + uw^T;
fim função pos_house
```

% Este algoritmo envolve  $4mn$  flops

## Gauss, Givens ou Householder?

O algoritmo **gauss** que permite construir a matriz de Gauss  $M_1 \in \mathbb{R}^{n \times n}$  requer  $n - 1$  operações, enquanto o algoritmo **house** para “construir” a matriz de Householder requer  $5n$  operações. Para reduzir um vector  $a \in \mathbb{R}^n$  à forma  $ke_1$  são necessárias, em geral,  $n - 1$  matrizes de Givens, pelo que este processo requer, em geral,  $(n - 1) \times 6$  operações.

A pré-multiplicação de uma matriz  $A$  por  $M_1, R_{1n} \cdots R_{12}$  e  $P$  envolve  $2n^2 - 2n, 6n^2 - 6n$  e  $4n^2$  operações, respectivamente. As matrizes de Gauss são, por isso, as menos dispendiosas e as matrizes de Givens são, em geral, as que exigem um maior esforço computacional. (Convém notar que, com o método de Givens, se algumas das componentes do vector  $a$  são nulas, o número de operações será reduzido, uma vez que serão omitidas algumas transformações.)

No entanto, as matrizes unitárias (Givens e Householder) apresentam duas propriedades importantes que as colocam em muitas situações em vantagem em relação às matrizes não unitárias de Gauss.

1. Se  $A$  é hermitiana (real simétrica) e utilizarmos matrizes unitárias (ortogonais) numa transformação de semelhança, então a matriz resultante será ainda hermitiana (simétrica).
2. Suponhamos que  $B$  é unitariamente semelhante a  $A$ , ou seja que  $B = RAR^*$ , com  $R$  unitária. Então:

– Se  $x_1$  é vector próprio à direita de  $A$ , então  $Rx_1$  é vector próprio à direita de  $B$ . De igual modo, se  $y_1$  é vector próprio à esquerda de  $A$ , então  $R^T y_1$  é vector próprio à esquerda de  $B$ .

– Temos:

$$\begin{aligned} \|x_1\| = 1 &\Rightarrow \|Rx_1\| = 1 \\ \|y_1\| = 1 &\Rightarrow \|R^T y_1\| = 1. \end{aligned}$$

– Além disso

$$|(R^T y_1)^T (R^* x_1)| = |y_1^T x_1|.$$

Os resultados anteriores, juntamente com a desigualdade (1.12), mostram que a sensibilidade dos valores próprios de  $A$  a alterações nos seus elementos permanece invariante sob transformações de semelhança unitárias, ou seja, que não há deterioração da condição de  $A$  por uma transformação deste tipo. Tal resultado garante a estabilidade dos métodos que envolvem matrizes unitárias.

### 3. Métodos Iterativos

Como já vimos anteriormente, o problema de cálculo de valores próprios de uma matriz  $A$  pode ser reduzido ao problema de determinação dos zeros de um dado polinómio (o polinómio característico de  $A$ ). Reciprocamente, todo o problema de determinação dos zeros de um polinómio mónico  $p(z)$  pode ser formulado como um problema de cálculo de valores próprios de uma certa matriz (a matriz companheira de  $p$ ; veja Exercício 4 do Capítulo 1). Como é bem sabido, Galois provou que não existe nenhuma “fórmula resolvente” para polinómios de grau  $\geq 5$ . Este resultado implica que, mesmo trabalhando com aritmética exacta, não é possível escrever um programa que calcule os zeros de um polinómio qualquer, num número finito de passos. Esta mesma conclusão se aplica ao problema mais geral de cálculo dos valores próprios de uma matriz, i.e. *qualquer método para obtenção dos valores próprios de uma dada matriz tem que ser iterativo*.

Os métodos numéricos para calcular os valores e vectores próprios de uma matriz, embora sendo todos de carácter iterativo, são vulgarmente agrupados em duas classes – a dos chamados *métodos iterativos* e a dos *métodos de transformação*. O que caracteriza os métodos de transformação (também ditos métodos directos) é a utilização de transformações de semelhança para reduzir (sucessivamente) a matriz inicial a uma outra que possua uma forma mais adequada ao cálculo dos valores e vectores próprios.

Neste capítulo, faremos o estudo de alguns métodos iterativos. Os métodos aqui estudados têm como principal finalidade a determinação de um (ou alguns) vector(es) próprio(s), e são especialmente adequados ao caso de matrizes dispersas de grande dimensão, para as quais sejam conhecidas boas aproximações iniciais para os vectores próprios.

#### 3.1 Método da potência

**Definição 3.1** *Seja  $A \in M_n$  com valores próprios  $\lambda_1, \lambda_2, \dots, \lambda_n$  tais que*

$$|\lambda_1| = |\lambda_2| = \dots = |\lambda_r| > |\lambda_{r+1}| \geq \dots \geq |\lambda_n|. \quad (3.1)$$

Nestas condições, diremos que os valores próprios  $\lambda_1, \lambda_2, \dots, \lambda_r$  são **dominantes**, sendo os respectivos vectores próprios associados designados por vectores próprios dominantes.

Consideremos o seguinte esquema iterativo, que constitui o chamado

**Método da Potência:**

$$\left. \begin{aligned} y_k &= \frac{1}{\gamma_k} A y_{k-1}; \quad k = 1, 2, 3, \dots, \\ y_0 &\neq 0, \end{aligned} \right\} \quad (3.2)$$

onde  $\gamma_k$  é o elemento de maior módulo do vector  $A y_{k-1}$ .

O teorema seguinte estabelece condições sob as quais a sequência de iterações definida por (3.2) converge.

**Teorema 3.1** *Se  $A$  é não-deficiente e  $|\lambda_1| > |\lambda_i|$ ;  $i = 2, \dots, n$ , então  $\{y_k\}$  converge para um vector próprio associado ao valor próprio dominante  $\lambda_1$  e  $\{\gamma_k\}$  converge para  $\lambda_1$ .*

**Dem:** Segue-se de (3.2) que

$$y_k = \frac{1}{\gamma_k} A y_{k-1} = \dots = \frac{1}{\left(\prod_{i=1}^k \gamma_i\right)} A^k y_0 = \frac{1}{c_k} A^k y_0, \quad (3.3)$$

onde

$$c_k = \prod_{i=1}^k \gamma_i. \quad (3.4)$$

Como, por hipótese,  $A$  é não-deficiente, ou seja, existem  $n$  vectores próprios linearmente independentes  $x_i$ ;  $i = 1, \dots, n$ , o vector  $y_0$  admite uma expansão da forma

$$y_0 = \sum_{i=1}^n \alpha_i x_i, \quad (3.5)$$

onde os escalares  $\alpha_i$  não são todos nulos. Suponhamos que  $\alpha_1 \neq 0$ . De (3.3)–(3.5) vem:

$$\begin{aligned} y_k &= \frac{1}{c_k} A^k \left\{ \sum_{i=1}^n \alpha_i x_i \right\} \\ &= \frac{1}{c_k} \left\{ \sum_{i=1}^n \alpha_i \lambda_i^k x_i \right\} \\ &= \frac{\lambda_1^k}{c_k} \left\{ \alpha_1 x_1 + \sum_{i=2}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^k x_i \right\} \\ &\rightarrow \text{múltiplo de } x_1 \text{ quando } k \rightarrow \infty, \end{aligned}$$

uma vez que  $|\frac{\lambda_i}{\lambda_1}| < 1; i = 2, \dots, n$ .

Como  $y_k$  converge para um vector próprio associado ao valor próprio  $\lambda_1$ , então, no limite,

$$Ay_k = \lambda_1 y_k.$$

Mas, o elemento de maior módulo do vector  $\lambda_1 y_k$  é  $\lambda_1$ , uma vez que  $y_k$  tem 1 como componente de maior módulo. Logo, temos

$$\gamma_{k+1} \rightarrow \lambda_1.$$

□

**Nota 3.1** A demonstração anterior assume que  $\alpha_1 \neq 0$ , ou seja, que o vector inicial tem componente não-nula na direcção de  $x_1$ . Se  $\alpha_1 = 0$  e  $|\lambda_2| > |\lambda_i|; i = 3, \dots, n$ , então, teoricamente, a iteração convergirá para um múltiplo de  $x_2$  e  $\lambda_2$ . Na prática, no entanto, os erros de arredondamento introduzem geralmente uma componente na direcção de  $x_1$  e portanto, eventualmente, a iteração convergirá para um múltiplo de  $x_1$  e  $\lambda_1$ . No entanto, uma boa aproximação para  $x_2$  e  $\lambda_2$  é por vezes obtida, antes do termo em  $\lambda_1$  dominar.

**Nota 3.2** Note-se que a rapidez de convergência depende da grandeza de  $\alpha_1$  comparada com os restantes valores de  $|\alpha_i|$  mas, sobretudo, das razões  $|\lambda_2/\lambda_1|, \dots, |\lambda_n/\lambda_1|$ , sendo tanto mais rápida quanto menores forem essas razões. Em particular, se  $|\lambda_2/\lambda_1| \approx 1$ , é natural que a convergência seja bastante lenta.

Suponhamos agora que, em vez de existir um único valor próprio dominante  $\lambda_1$ , existem  $r$  valores próprios dominantes *iguais*, i.e. temos

$$\lambda_1 = \lambda_2 = \dots = \lambda_r, \quad |\lambda_1| > |\lambda_i|; i = r+1, \dots, n.$$

Neste caso, ter-se-á

$$\begin{aligned} y_k &= \frac{\lambda_1^k}{c_k} \left\{ \sum_{i=1}^r \alpha_i x_i + \sum_{i=r+1}^n \left( \frac{\lambda_i}{\lambda_1} \right)^k \alpha_i x_i \right\} \\ &\rightarrow \frac{\lambda_1^k}{c_k} \left( \sum_{i=1}^r \alpha_i x_i \right) \text{ quando } k \rightarrow \infty. \end{aligned}$$

Como  $x_1, x_2, \dots, x_r$  são vectores próprios associados ao mesmo valor próprio  $\lambda_1 (= \lambda_2 = \dots = \lambda_r)$ ,  $\sum_{i=1}^r \alpha_i x_i$  é ainda um vector próprio associado a  $\lambda_1$ . Quer dizer, ainda neste caso, são válidas as conclusões do Teorema 3.1. No entanto, se existirem valores próprios dominantes

diferentes mas com o mesmo módulo, o processo iterativo falha, ou seja, as iterações sucessivas oscilam. Consideremos, por exemplo, o caso em que

$$\begin{aligned}\lambda_2 &= -\lambda_1, \\ |\lambda_1| &> |\lambda_i|; i = 3, \dots, n.\end{aligned}$$

Neste caso, temos

$$y_k = \frac{\lambda_1^k}{c_k} \{ \alpha_1 x_1 + (-1)^k \alpha_2 x_2 + \sum_{i=3}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^k x_i \}, \quad (3.6)$$

$$y_{k+1} = \frac{\lambda_1^{k+1}}{c_{k+1}} \{ \alpha_1 x_1 - (-1)^k \alpha_2 x_2 + \sum_{i=3}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^{k+1} x_i \} \quad (3.7)$$

e

$$\begin{aligned}y_{k+2} &= \frac{\lambda_1^{k+2}}{c_{k+2}} \{ \alpha_1 x_1 + (-1)^k \alpha_2 x_2 + \sum_{i=3}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^{k+2} x_i \} \\ &= \frac{\lambda_1^{k+2}}{\gamma_{k+2} \gamma_{k+1} c_k} \{ \alpha_1 x_1 + (-1)^k \alpha_2 x_2 + \sum_{i=3}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^{k+2} x_i \},\end{aligned} \quad (3.8)$$

o que mostra que os vectores oscilam. Além disso, segue-se de (3.6) e (3.8) que, no limite,

$$y_{k+2} = \frac{\lambda_1^2}{\gamma_{k+2} \gamma_{k+1}} y_k.$$

Como os vectores  $y_{k+2}$  e  $y_k$  têm ambos 1 como componente de maior módulo, vem que

$$\lim_{k \rightarrow \infty} \frac{\lambda_1^2}{\gamma_{k+2} \gamma_{k+1}} = 1,$$

isto é, que

$$\lim_{k \rightarrow \infty} \gamma_{k+2} \gamma_{k+1} = \lambda_1^2. \quad (3.9)$$

No caso de existirem dois valores próprios dominantes que sejam complexos conjugados,  $\lambda_2 = \overline{\lambda_1}$  podem determinar-se, no limite, coeficientes  $b$  e  $c$  tais que  $\lambda^2 + b\lambda + c = 0$  tem  $\lambda_1$  e  $\overline{\lambda_1}$  como raízes; veja, por exemplo, [Wil65, pp 579,580].



**Algoritmo 3.1 Método de Potência**

```

% Cálculo do valor e vector próprio dominante de uma matriz  $A$  de ordem  $n$ ,
% dada uma aproximação inicial  $y_0$  para o vector próprio
% Parâmetros de entrada: matriz  $A$ , aproximação inicial  $y_0$ 
% Parâmetros de saída: valor e vector próprio dominante  $\lambda_k$  e  $y_k$ 

```

```

função  $[y_k, \lambda_k] = \text{potencia}(A, y_0)$ 
     $k = 0$ ;
    while criterio_paragem falso
         $k = k + 1$ ;
         $\tilde{y} = Ay_{k-1}$ ;
         $\lambda_k = y_{max}$ ;      %  $|y_{max}| = \max_i |\tilde{y}_i|$ ;
         $y_k = \tilde{y}/\lambda_k$ ;
    end
fim função potencia

```

**Nota 3.3** O algoritmo anterior deve terminar de acordo com um critério de paragem (*criterio\_paragem*) a estabelecer pelo utilizador. Os critérios de paragem mais usuais são:

- $k = k_{max}$ ;
- $|\lambda_k - \lambda_{k-1}| < \epsilon$ ;
- $|y_k - y_{k-1}| < \delta$ ;

Note-se ainda que, em cada iteração  $k$ , só é necessário conhecer o valor de  $y_{k-1}$  e  $\lambda_{k-1}$ .

## 3.2 Aceleração de convergência

Como já referimos, a rapidez de convergência para um vector próprio associado a um valor próprio dominante depende, essencialmente, da razão  $|\lambda_2|/|\lambda_1|$ . Se considerarmos o método da potência aplicado à matriz  $A - pI$  em vez de  $A$ , então, uma vez que os valores próprios dessa matriz são  $\lambda_i - p$ , desde que  $\lambda_1 - p$  seja ainda dominante, a rapidez de convergência será "governada" pela razão  $|\lambda_2 - p|/|\lambda_1 - p|$ . Para uma escolha adequada de  $p$ , esta razão pode ser bastante menor do que  $|\lambda_2|/|\lambda_1|$ . A este método de acelerar a convergência, que consiste portanto na aplicação do método da potência a uma matriz  $A - pI$ , chamamos *translação de origem*. Para certas distribuições dos valores próprios, a translação de origem é um método bastante eficaz de acelerar a convergência.

Por exemplo, se  $A$  é uma matriz de ordem 4 com valores próprios  $\lambda_j = 15 - j$ ;  $j = 1, 2, 3, 4$ , a iteração normal convergirá para  $x_1$  com rapidez governada essencialmente pela razão  $(13/14)^k$ ;

se escolhermos  $p = 12$  e trabalharmos com  $(A - pI)$  a convergência será governada pela razão  $(1/2)^k$ , sendo portanto muito mais rápida.

Embora o valor de  $p$  possa, por vezes, ser escolhido adequadamente, caso, por exemplo, haja algum conhecimento sobre a distribuição dos valores próprios, é difícil estabelecer um processo automático para a sua escolha, fazendo-se esta, por vezes, por tentativas.

Outro processo simples de acelerar a convergência para o valor próprio dominante, usado para matrizes reais simétricas, baseia-se na utilização do quociente de Rayleigh e pode ser descrito do seguinte modo: <sup>1</sup>

Se  $A$  é real, simétrica, então a base de vectores próprios  $\{x_i\}$  referida na demonstração do Teorema 3.1 pode ser escolhida como ortonormada, i.e. satisfazendo

$$x_i^T x_j = \delta_{ij}. \quad (3.10)$$

Consideremos o vector  $y_k$  obtido na iteração  $k$  do método da potência e formemos o quociente de Rayleigh para esse vector,

$$\frac{y_k^T A y_k}{y_k^T y_k} \quad (3.11)$$

Então, usando (3.5) e atendendo a (3.10), vem

$$\frac{y_k^T A y_k}{y_k^T y_k} = \frac{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2k}} = \lambda_1 + \mathcal{O}\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right). \quad (3.12)$$

Em geral, o quociente de Rayleigh correspondente a  $y_k$  dará uma melhor aproximação para  $\lambda_1$  do que a fornecida pelo método da potência.

**Nota 3.4** Note-se que

$$\frac{y_k^T A y_k}{y_k^T y_k} = \frac{y_k^T \tilde{y}_{k+1}}{y_k^T y_k}$$

onde  $\tilde{y}_{k+1}$  denota o vector obtido na próxima iteração do método da potência, antes de ser normalizado.

**Nota 3.5** Sob o ponto de vista de aplicação prática, o método da potência tem algumas características atraentes e algumas desvantagens. A sua principal vantagem é a forma extremamente simples da recursão (3.2). Por exemplo, para efectuarmos o produto  $A y_k$  apenas intervêm os elementos não-nulos de  $A$ ; assim, este método é especialmente adequado ao caso em que  $A$  é uma matriz dispersa.

---

<sup>1</sup>O método é descrito para matrizes reais simétricas, mas tem uma generalização óbvia para matrizes hermitianas.

A principal desvantagem é que a convergência é lenta se  $\lambda_1$  não for fortemente dominante. Embora, como vimos, a convergência possa ser acelerada (por vezes de forma significativa) pelo uso de translações de origem, a escolha dos valores de  $p$  não se presta a computação automática.

### 3.3 Deflação

Suponhamos que os valores próprios  $\lambda_1, \lambda_2, \dots, \lambda_n$  da matriz  $A$  são tais que  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ , que  $\lambda_1$  e  $x_1$  foram já determinados e que se pretende agora determinar os valores próprios  $\lambda_2, \dots, \lambda_n$  (chamados valores próprios subdominantes) e respectivos vectores próprios. É natural perguntar se o conhecimento de  $\lambda_1$  e  $x_1$  pode ser usado para garantir a convergência do método da potência para um vector próprio diferente de  $x_1$ , sem risco de se convergir novamente para  $x_1$ . Chama-se *deflação* a qualquer método que consista em, conhecidos  $\lambda_1$  e  $x_1$ , determinar, a partir de  $A$ , uma matriz  $B$  cujos valores próprios sejam essencialmente os restantes valores próprios da matriz original. Uma aplicação repetida deste processo de deflação permite-nos determinar sequencialmente os valores próprios e vectores próprios subdominantes. Existem vários processos de deflação, dos quais, os mais populares, são baseados em transformações de semelhança.

Suponhamos então que o valor próprio dominante  $\lambda_1$  e vector próprio associado  $x_1$  de  $A_1 := A$  foram calculados. Seja  $R_1$  uma matriz não-singular tal que

$$R_1 x_1 = k e_1, \quad (3.13)$$

com  $k \neq 0$ . (A maneira como esta matriz  $R_1$  pode ser determinada foi já discutida na Secção 2.2.) Então,

$$R_1 A_1 (R_1^{-1} R_1) x_1 = \lambda_1 R_1 x_1,$$

donde,

$$(R_1 A_1 R_1^{-1}) e_1 = \lambda_1 e_1.$$

Podemos assim escrever

$$A_2 = R_1 A_1 R_1^{-1} = \left( \begin{array}{c|c} \lambda_1 & x^T \\ \hline O & B_2 \end{array} \right),$$

onde  $B_2$  é uma matriz de ordem  $(n - 1)$  e  $x$  um vector de  $n - 1$  elementos. Como  $A_2$  tem os mesmos valores próprios que  $A_1$ , segue-se que os valores próprios de  $B_2$  são  $\lambda_2, \lambda_3, \dots, \lambda_n$ . Assim, podemos trabalhar com  $B_2$  para obter o próximo valor próprio  $\lambda_2$  e correspondente vector próprio  $y_2$  satisfazendo

$$B_2 y_2 = \lambda_2 y_2. \quad (3.14)$$

Resta-nos determinar o vector próprio  $x_2$  de  $A_1$  associado a  $\lambda_2$ . Seja  $z_2$  o vector próprio de  $A_2$  associado a  $\lambda_2$ . Então,

$$\left( \begin{array}{c|c} \lambda_1 & x^T \\ \hline O & B_2 \end{array} \right) z_2 = \lambda_2 z_2, \quad (3.15)$$

e, usando (3.15) podemos tomar

$$z_2 = \begin{pmatrix} \alpha \\ \hline y_2 \end{pmatrix}, \quad (3.16)$$

com  $\alpha$  um escalar satisfazendo

$$(\lambda_1 - \lambda_2)\alpha + x^T y_2 = 0. \quad (3.17)$$

Finalmente, calculando  $z_2$ , tem-se, como já vimos,

$$x_2 = R_1^{-1} z_2. \quad (3.18)$$

Continuando este processo, podemos obter os restantes valores próprios e vectores próprios subdominantes.

**Nota 3.6** Note-se que sucessivas deflações darão, no limite, uma redução de semelhança à forma triangular superior.

### 3.4 Método da potência inverso

Os métodos para determinar alguns dos valores próprios dominantes/subdominantes que descrevemos até agora convergem, em geral, lentamente. Nesta secção descreveremos um método, chamado *método da potência inverso*, o qual, é um método de convergência bastante rápida.

Seja  $p$  um escalar, diferente de qualquer valor próprio de  $A$ . Se

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

são os valores próprios de  $A$ , então

$$\frac{1}{\lambda_1 - p}, \frac{1}{\lambda_2 - p}, \dots, \frac{1}{\lambda_n - p}$$

são os valores próprios de  $(A - pI)^{-1}$ . Se aplicarmos o método da potência à matriz  $(A - pI)^{-1}$ , vamos obter o valor próprio de maior módulo desta matriz, isto é,  $\frac{1}{\lambda_j - p}$  onde  $|\lambda_j - p| < |\lambda_i - p|$

$p|; i \neq j$ . Por outras palavras, aplicando o método da potência à matriz  $(A - pI)^{-1}$ , obter-se-á o valor próprio (e correspondente vector próprio) que, no plano complexo, está mais próximo de  $p$ . Note-se que, se

$$\beta_j = \frac{1}{\lambda_j - p}, \quad (3.19)$$

então,

$$\lambda_j = \frac{1}{\beta_j} + p. \quad (3.20)$$

Chama-se *método da potência inverso* ao método que acabámos de descrever, ou seja, ao método da potência aplicado a uma matriz do tipo  $(A - pI)^{-1}$ . Assim, (cf. Eq.(3.2)), o método da potência inverso é definido pelo seguinte esquema iterativo:

$$y_k = \frac{1}{\gamma_k} (A - pI)^{-1} y_{k-1}; \quad k = 1, 2, 3, \dots \quad (3.21)$$

com

$$y_0 \neq 0,$$

onde  $\gamma_k$  é a componente de maior módulo do vector  $(A - pI)^{-1} y_{k-1}$ . Para maior eficiência, a fórmula iterativa (3.21) é reescrita como

$$\left. \begin{aligned} (A - pI) \tilde{y}_k &= y_{k-1}, \\ y_k &= \frac{1}{\gamma_k} \tilde{y}_k; \quad k = 1, 2, 3, \dots \end{aligned} \right\} \quad (3.22)$$

onde  $\gamma_k$  é a componente de maior módulo de  $\tilde{y}_k$ . (Recorde-se que a quantidade de cálculo envolvida na resolução de um sistema de equações, é bastante menor do que a envolvida na determinação da inversa de uma matriz.)

Note-se que, ao aplicar o processo iterativo (3.22), teremos de resolver sucessivos sistemas de equações, os quais possuem todos a mesma matriz de coeficientes,  $A - pI$ . Assim, um método aconselhável para a resolução destes sistemas será utilizando a decomposição LU.<sup>2</sup>

Por outras palavras, uma vez efectuada a decomposição

$$A - pI = LU \quad (3.23)$$

proceder-se-á da seguinte forma:

$$\left. \begin{aligned} Lz &= y_{k-1}, \\ U \tilde{y}_k &= z, \\ y_k &= \frac{1}{\gamma_k} \tilde{y}_k; \quad k = 1, 2, 3, \dots \end{aligned} \right\} \quad (3.24)$$

---

<sup>2</sup>Para garantia de estabilidade, a decomposição LU deve ser efectuada com escolha parcial de pivot. No entanto, e por uma questão de simplicidade, descreveremos o método omitindo tal escolha.

Prova-se que a escolha do vector inicial  $y_0$  não é particularmente importante; veja, por exemplo, [GW73, pp 57,58]. O procedimento usual é omitir o primeiro passo em (3.24) quando  $k = 1$  e obter  $\tilde{y}_1$  através da relação

$$U\tilde{y}_1 = e$$

onde  $e^T = (1, 1, \dots, 1)$ . (Isto é, evidentemente, equivalente a tomar  $y_0 = Le$ .)

A discussão que efectuámos sobre o método da potência, mostra imediatamente que, se  $A$  é não-deficiente e se  $|\lambda_j - p| < |\lambda_i - p|$ ;  $i \neq j$ , então o método da potência inverso irá convergir para um vector próprio associado a  $\lambda_j$  (e dando, também, como é claro, uma sequência de valores convergindo para  $\lambda_j$ ). Pode provar-se que esta convergência é, em geral, muito rápida, sobretudo se  $p$  for uma aproximação razoável para o valor próprio  $\lambda_j$  e se este valor próprio for bastante *isolado* dos restantes. De facto, na prática, com  $p \approx \lambda_j$ , é usual efectuar apenas um ou dois passos deste método para obter uma boa aproximação para o vector próprio associado a  $\lambda_j$ .

Em resumo, podemos dizer que o método da potência inverso é um dos processos mais poderosos de determinação de um vector próprio associado a um valor próprio do qual é conhecida uma boa aproximação.

**Nota 3.7** Embora  $A - pI$  seja uma matriz mal condicionada a respeito da inversão – basta notar que se  $p$  é muito próximo do valor próprio  $\lambda_j$  então  $A - pI$  é "quase" singular – pode provar-se que o método da potência inverso é um processo estável de determinação do vector próprio associado a  $\lambda_j$ ; com efeito, todas as componentes do vector solução de (3.22), embora possivelmente muito pouco correctas, têm erros na mesma proporção, ou seja,  $y_k$  é "quase" proporcional a  $x_j$ , o que não afecta os resultados finais. Este resultado pode provar-se rigorosamente considerando a análise de erros para a solução de equações lineares por eliminação Gaussiana; veja, por exemplo, [Wil65, pp 620,621] e [?, pp226,227].

**Nota 3.8** Uma aparente fraqueza o método da potência inverso na determinação de vectores próprios é que, se  $\lambda_j$  é um valor próprio múltiplo, determinar-se-á apenas *um* vector próprio correspondente. Contudo, consegue-se geralmente determinar os diferentes vectores próprios associados a  $\lambda_j$ , perturbando ligeiramente  $\lambda_j$ . Em alternativa, a utilização de diferentes escolhas do vector inicial  $y_0$  conduzirá geralmente a vectores próprios linearmente independentes.

### Algoritmo 3.2 Método de Potência Inverso

% Cálculo do valor e vector próprio dominante de uma matriz  $A$  de ordem  $n$ ,  
% dada uma translação  $p$   
% **Parâmetros de entrada:** matriz  $A$ , translação  $p$   
% **Parâmetros de saída:** valor e vector próprio dominante  $\lambda_k$  e  $y_k$

```
função [yk, λk]=potinverso(A, p)
    k = 0;
    A = A - pI;
    [L U]=lu(A);    % função lu do MATLAB
    y0 = Le;    % e = (1...1)T
    while critério_paragem falso
        k = k + 1;
        ã = resolve_sistema(L, U, yk-1);
        λk = ymax;    % |ymax| = maxi |ãi|;
        yk = ã/λk;
    end
    λk = 1/λk + p;
fim função potinverso
```

**Nota 3.9** A função **resolve\_sistema** deve indicar a solução do sistema  $LU\tilde{y} = y_{k-1}$  ou  $PLU\tilde{y} = y_{k-1}$ , caso a decomposição LU da matriz tenha sido obtida com escolha parcial de pivot. (Neste caso, a função **lu** do MATLAB deve ser usada com 3 parâmetros de saída:  $[L \ U \ P] = lu(A)$ .)

## 3.5 Notas e referências adicionais

- Métodos que generalizam o método da potência para calcular vários vectores próprios simultaneamente, foram introduzidos por [?] , sob os nomes de *iteração em escada* e de *bi-iteração* . Wilkinson [Wil65] descreve uma variante desses métodos usando matrizes ortogonais. Para matrizes simétricas, esta variante pode ser acelerada por um processo descoberto independentemente por Clint e Jennings [?]), Stewart [Ste69] e Rutishauser [?].
- A iteração do quociente de Rayleigh e suas variantes foram analisadas exaustivamente numa série de artigos da autoria de Ostrowski [?] .
- Outros processos de fazer deflação são descritos em [Wil65] e [?] .
- O método da potência inverso é devido a Wielandt [?] . A implementação prática deste método é tratada em [Wil65] e em [?] .

### 3.6 Exercícios



1. Considere a matriz

$$A = \begin{pmatrix} -3 & 1 & 0 \\ 1 & -3 & -3 \\ 0 & -3 & 4 \end{pmatrix}.$$

- a) Faça 5 iterações do Método da Potência, para obter uma aproximação para o valor próprio dominante de  $A$ .
- b) Faça 2 iterações do Método da Potência, para obter uma aproximação para o valor próprio dominante de  $A + 4I$ .
- c) Comente a rapidez de convergência do método nos dois casos.  
(Use em a) e b) como aproximação inicial  $y_0 = (0 \ 0 \ 1)^T$ .)



2. Considere a matriz

$$A = \begin{pmatrix} 13 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{37}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{61}{5} \end{pmatrix}.$$

- a) Faça 6 iterações do Método da Potência para obter uma aproximação para o valor próprio dominante e respectivo vector próprio, da matriz  $A$ , tomando como aproximação inicial o vector  $x = (1 \ 1 \ 1)^T$ .
  - b) Comente o resultado obtido na alínea anterior.  
(Pode recorrer à função *eig* do *MATLAB*.)
  - c) Que pode dizer relativamente à rapidez de convergência do Método da Potência aplicado às matrizes  $A_1 = A - 13I$  e  $A_2 = A - 12I$ ?
3. Suponha que os valores próprios de uma matriz quadrada  $A$ , de ordem  $n$ , estão ordenados da seguinte maneira:

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \cdots \geq |\lambda_n|$$

e que a matriz  $A$  tem vectores próprios linearmente independentes  $v_1, v_2, \dots, v_n$ .

- a) Diga para que vector próprio e valor próprio convergirá o método da potência se for usado o vector inicial

$$x_0 = \alpha_2 v_2 + \alpha_3 v_3 + \cdots + \alpha_n v_n$$



b) Mostre que dado um vector  $x$  qualquer ( $x = \sum_{i=1}^n \alpha_i v_i$ ), o vector  $x_0 = (A - \lambda_1 I)x$  está nas condições referidas em a).

4. Sejam  $\lambda_1 < \lambda_2 < \dots < \lambda_n$  os valores próprios de uma matriz  $A$  de ordem  $n$ . Para cada um dos casos seguintes, descreva o comportamento do Método da Potência aplicado à matriz  $(A - pI)^{-1}$ , onde:

a)  $p = \lambda_2$ .

b)  $p = (\lambda_1 + \lambda_2)/2$ .

c)  $p = \lambda_1 - \varepsilon$ ,  $\varepsilon > 0$ .



5. Considere a matriz

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 5 & 0 \\ 2 & 0 & 3 \end{pmatrix}.$$

a) Faça 6 iterações do Método da Potência para calcular uma aproximação para um valor próprio e respectivo vector próprio da matriz  $A$ .

(Tome como aproximação inicial o vector  $x = (-1 \ 1 \ 0)^T$ .)

b) Justifique o comportamento do método da potência, neste caso.

(Pode recorrer à função *eig* do *MATLAB*.)

c) Sem realizar qualquer iteração, diga, justificando, para que valor próprio de  $A$  converge o Método da Potência Inverso aplicado às matrizes  $A_1 = A - 8I$  e  $A_2 = A - 3I$ .



6. Os valores próprios da matriz

$$A = \begin{pmatrix} 10 & -3 & 6 \\ -3 & 3 & -2 \\ 6 & -2 & 1 \end{pmatrix}$$

são (com 4 casas decimais)  $\lambda_1 = 14.1493$ ,  $\lambda_2 = -2.0477$ ,  $\lambda_3 = 1.8983$

a) Use o Método da Potência para calcular, com uma casa decimal correcta, o valor próprio dominante e o respectivo vector próprio.

b) Calcule o quociente de Rayleigh

$$\frac{y_k^T A y_k}{y_k^T y_k}$$

onde  $y_k$  é a aproximação obtida em a) para o vector próprio dominante. Compare este valor com o valor de  $\lambda_1$ . O que conclui ?

- c) Se usar o método da potência com a matriz  $(A - 1.89I)^{-1}$  para que valor (e vector) converge o método?
- d) Obtenha a decomposição  $LU$  da matriz  $A - 1.89I$ .
- e) Faça duas iterações do Método da Potência Inverso, usando o vector inicial  $x^{(0)} = (0.32, 0.94, 0.03)^T$ . Indique a aproximação obtida para o valor próprio de  $A$ .



7. Use o Método da Potência Inverso para determinar uma aproximação, com 5 casas decimais correctas, para o valor próprio mais próximo de 6, da matriz

$$A = \begin{pmatrix} 6 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Indique também uma aproximação para o vector próprio associado ao valor próprio calculado.

8. Considere a matriz

$$A = \begin{pmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{pmatrix}.$$

Suponha que  $A$  tem  $\lambda_1 = 11.0$  como valor próprio dominante, com vector próprio associado  $x_1 = (0.5, 1.0, 0.75)^T$ . Determine os restantes valores próprios de  $A$ , usando técnica de deflação .

## 4. Métodos de Transformação

No capítulo anterior foram descritos algoritmos que se destinam a calcular um único valor próprio e correspondente vector próprio de cada vez. Foi também explicado como, usando deflação, poderíamos repetir o processo por forma a obter os diversos valores próprios de uma matriz  $A$ . Neste capítulo serão descritos algoritmos cuja finalidade é o cálculo de todos os valores próprios de  $A$  de uma só vez. A base destes algoritmos será o uso de transformações de semelhança para reduzir a matriz inicial a uma matriz cuja forma permita facilmente o cálculo dos seus valores e vectores próprios.

### 4.1 Método de Jacobi

O chamado *método de Jacobi clássico* que introduzimos nesta secção é aqui descrito apenas para matrizes reais simétricas, podendo, no entanto, ser generalizado facilmente para matrizes hermitianas. Sabemos já que, se  $A$  é hermitiana, então existe uma matriz unitária  $P$  tal que

$$PAP^* = \text{diag}(\lambda_i) \quad (4.1)$$

(veja a demonstração da Propriedade P2., p.9. Se  $A$  é real e simétrica, a matriz  $P$  em (4.1) pode ser escolhida como real e, portanto, ortogonal. A ideia do *método de Jacobi clássico* é tentar construir a matriz  $P$  usando sucessivas transformações de rotação plana. (Relembre que uma matriz  $R_{ij}$  de rotação plana é uma matriz ortogonal.) Na realidade, o número de rotações planas necessárias para a redução de  $A$  à forma diagonal é, em geral, infinito. Na prática, o processo pára quando os elementos fora da diagonal forem negligíveis comparados com os da diagonal. A fórmula iterativa seguinte poderá descrever o

**Método de Jacobi Clássico:**

$$\left. \begin{aligned} A_{k+1} &= R_k A_k R_k^T; \quad k = 1, 2, 3, \dots, \\ A_1 &= A, \end{aligned} \right\} \quad (4.2)$$

com  $R_k \equiv R_{ij}^{(k)}(\theta)$  uma matriz de rotação plana escolhida de forma a anular  $a_{ij}^{(k)}$ , onde  $a_{ij}^{(k)}$  é o elemento não-diagonal de maior módulo da matriz  $A_k$ , i.e.

$$|a_{ij}^{(k)}| = \max_{p \neq q} |a_{pq}^{(k)}|.$$

Note-se que  $A_{k+1}$  tem os mesmos elementos que  $A_k$  excepto nas linhas  $i$  e  $j$ , onde temos:

$$\left. \begin{aligned} a_{il}^{(k+1)} &= a_{il}^{(k+1)} = a_{il}^{(k)} \cos \theta + a_{jl}^{(k)} \sin \theta & l \neq i, j, \\ a_{jl}^{(k+1)} &= a_{jl}^{(k+1)} = -a_{il}^{(k)} \sin \theta + a_{jl}^{(k)} \cos \theta & l \neq i, j, \\ a_{ii}^{(k+1)} &= a_{ii}^{(k)} \cos^2 \theta + 2a_{ij}^{(k)} \sin \theta \cos \theta + a_{jj}^{(k)} \sin^2 \theta, \\ a_{jj}^{(k+1)} &= a_{ii}^{(k)} \sin^2 \theta - 2a_{ij}^{(k)} \sin \theta \cos \theta + a_{jj}^{(k)} \cos^2 \theta, \\ a_{ij}^{(k+1)} &= a_{ji}^{(k+1)} = a_{ij}^{(k)} (\cos^2 \theta - \sin^2 \theta) + (a_{jj}^{(k)} - a_{ii}^{(k)}) \sin \theta \cos \theta. \end{aligned} \right\} \quad (4.3)$$

O ângulo de rotação  $\theta$  é escolhido de forma a anular  $a_{ij}^{(k+1)}$  i.e. é tal que

$$a_{ij}^{(k)} (\cos^2 \theta - \sin^2 \theta) + (a_{jj}^{(k)} - a_{ii}^{(k)}) \sin \theta \cos \theta = 0,$$

ou seja, tal que

$$a_{ij}^{(k)} \cos 2\theta + (a_{jj}^{(k)} - a_{ii}^{(k)}) \sin \theta \cos \theta = 0,$$

ou ainda, tal que

$$\alpha := \tan(2\theta) = \frac{2a_{ij}^{(k)}}{a_{ii}^{(k)} - a_{jj}^{(k)}}. \quad (4.4)$$

O valor de  $\theta$  é sempre escolhido satisfazendo

$$-\pi/4 \leq \theta \leq \pi/4. \quad (4.5)$$

Se  $a_{ii}^{(k)} = a_{jj}^{(k)}$ , escolhe-se  $\theta$  com o mesmo sinal de  $a_{ij}^{(k)}$ , isto é,  $\theta = \frac{a_{ij}^{(k)}}{|a_{ij}^{(k)}|} \frac{\pi}{4}$ . De (4.4), podemos calcular os valores de  $\sin \theta$  e  $\cos \theta$ , através das fórmulas

$$\cos^2 \theta = \frac{1}{2} + \frac{1}{2\sqrt{\alpha^2 + 1}} \quad \text{e} \quad \sin^2 \theta = \frac{1}{2} - \frac{1}{2\sqrt{\alpha^2 + 1}},$$

obtendo-se então de (4.3) os elementos  $a_{pq}^{(k+1)}$ . O uso de fórmulas numericamente mais estáveis para a determinação do ângulo de rotação  $\theta$  e dos valores de  $a_{pq}^{(k+1)}$  é descrito por Rutishauser em [WR71]; veja também [SB80, pp 343,344].

Convém salientar que um elemento que é reduzido a zero num passo do método de Jacobi é, em geral, tornado não-nulo em passos seguintes, pelo que este método é essencialmente de carácter iterativo. Vamos mostrar que este método iterativo converge. Mais precisamente, temos:

**Teorema 4.1** A sequência de matrizes  $\{A_k\}$  definida pelo método de Jacobi clássico converge para uma matriz diagonal.

**Dem:** Designemos por  $S(A_k)$  a soma dos quadrados dos elementos não-diagonais de  $A_k$  i.e.,

$$S(A_k) = \sum_{p \neq q} \{a_{pq}^{(k)}\}^2.$$

Vamos começar por demonstrar que  $\{S(A_k)\}$  é uma sucessão monótona decrescente convergente para zero. Temos que

$$a_{pq}^{(k+1)} = a_{pq}^{(k)}; \quad p, q \neq i, j.$$

Além disso,

$$\begin{aligned} \{a_{il}^{(k+1)}\}^2 + \{a_{jl}^{(k+1)}\}^2 &= \{a_{il}^{(k)} \cos \theta + a_{jl}^{(k)} \sin \theta\}^2 + \{-a_{il}^{(k)} \sin \theta + a_{jl}^{(k)} \cos \theta\}^2 \\ &= \{a_{il}^{(k)}\}^2 \cos^2 \theta + \{a_{jl}^{(k)}\}^2 \cos^2 \theta \\ &= (\{a_{il}^{(k)}\}^2 + \{a_{jl}^{(k)}\}^2)(\sin^2 \theta + \cos^2 \theta) \\ &= \{a_{il}^{(k)}\}^2 + \{a_{jl}^{(k)}\}^2; \quad l \neq i, j. \end{aligned}$$

Temos então, claramente

$$S(A_{k+1}) - 2\{a_{ij}^{(k+1)}\}^2 = S(A_k) - 2\{a_{ij}^{(k)}\}^2.$$

Mas, sendo o ângulo de rotação  $\theta$  escolhido por forma que  $a_{ij}^{(k+1)} = 0$  vem

$$\begin{aligned} S(A_{k+1}) &= S(A_k) - 2\{a_{ij}^{(k)}\}^2 \\ &= \left(1 - \frac{2\{a_{ij}^{(k)}\}^2}{\sum_{p \neq q} \{a_{p,q}^{(k)}\}^2}\right) S(A_k) \\ &= \left(1 - \frac{2}{\frac{\sum_{p \neq q} \{a_{p,q}^{(k)}\}^2}{\{a_{ij}^{(k)}\}^2}}\right) S(A_k). \end{aligned}$$

Como  $|a_{ij}^{(k)}| = \max_{p \neq q} |a_{pq}^{(k)}|$ , temos que

$$\begin{aligned} \frac{\sum_{p \neq q} \{a_{p,q}^{(k)}\}^2}{\{a_{ij}^{(k)}\}^2} &\leq \text{n}^\circ \text{elem. fora da diagonal} \times 1 \\ &= n^2 - n, \end{aligned}$$

i.e., vem

$$\begin{aligned} S(A_{k+1}) &\leq \left(1 - \frac{2}{n^2 - n}\right) S(A_k) \\ &\leq \left(1 - \frac{2}{n^2 - n}\right)^k S(A_1). \end{aligned}$$

Mas, com  $n > 2$ ,  $|1 - \frac{2}{n^2 - n}| < 1$ , donde

$$\lim_{k \rightarrow \infty} S(A_{k+1}) = 0.$$

Vemos assim que os elementos não-diagonais da sequência de matrizes  $\{A_k\}$  tendem para zero. Para concluir a demonstração teríamos que mostrar que  $\{A_k\}$  converge para uma matriz diagonal fixa, ou seja, que

$$\lim_{k \rightarrow \infty} |a_{ii}^{(k+1)} - a_{ii}^{(k)}| = 0.$$

Tal resultado demonstra-se facilmente, usando as relações (4.3), com alguma manipulação trigonométrica e sabendo que  $\theta$  é tal que  $|\theta| \leq \pi/4$ ; veja, por exemplo, [GW73, p. 66] e [Wil65, p. 268].

No método de Jacobi clássico, o processo de procura do elemento não-diagonal de maior módulo requer  $N = \frac{1}{2}(n^2 - n)$  operações de comparação, enquanto a rotação de Jacobi propriamente dita requer  $4n$  multiplicações. O trabalho exigido no processo de procura é assim bastante grande comparado com o da própria transformação, sobretudo para  $n$  elevado, pelo que será preferível seleccionar o elemento a anular de uma forma mais simples. No chamado *método de Jacobi cíclico especial* não se efectuam quaisquer comparações, anulando-se sucessivamente os elementos nas posições  $(1, 2), (1, 3), \dots, (1, n); (2, 3), \dots, (2, n); \dots; (n - 1, n)$ , voltando-se novamente ao elemento  $(1, 2)$  etc. Forsythe e Henrici [FH61] mostraram que também esta versão do método de Jacobi converge para uma matriz diagonal, mas a demonstração deste resultado é bastante menos simples que a do Teorema 4.1.

Noutra variante do método de Jacobi, a que poderemos chamar *método de Jacobi do limiar* (em Inglês, "threshold Jacobi method"), efectua-se uma procura limitada do elemento a anular. Mais precisamente é estabelecido um *valor limiar* e efectua-se uma procura até que seja encontrado um elemento de módulo superior ao limiar. Este elemento é então anulado e o processo repetido até que todos os elementos não-diagonais sejam inferiores ao limiar. Como  $S(A_k) \rightarrow 0$  segue-se que o nível do limiar terá de ser reduzido à medida que o algoritmo prossegue. Eventualmente será exigido um limiar próximo do menor número representável no computador, considerando-se então que se obteve convergência.

Pode mostrar-se que, se os valores próprios de  $A$  são simples, a convergência para a forma diagonal da sequência de matrizes obtidas por aplicação de qualquer das versões do método de Jacobi é quadrática, no sentido em que

$$S(A_{k+N}) \leq K S(A_k)^2, \quad (4.6)$$

onde  $K$  é uma constante que depende da separação dos valores próprios e da ordem  $n$  da matriz  $A$  e onde  $N = \frac{1}{2}(n^2 - n)$ ; veja [Sch61] e [Sch64]. Um resultado semelhante a (4.6) é também estabelecido em [Sch61], mesmo quando os valores próprios de  $A$  são duplos.

Aplicando o método de Jacobi, vamos obter uma matriz que, dentro de uma certa precisão, tomamos como diagonal. Assim, temos

$$A_k = V_k A V_k^T \approx D, \quad (4.7)$$

onde  $D$  é uma matriz diagonal e  $V_k = R_k R_{k-1} \dots R_1$  é uma matriz ortogonal. Segue-se de (4.7) que os elementos diagonais de  $A_k$  são aproximações para os valores próprios  $\lambda_j$  de  $A$  e que as colunas de  $V_k$  são aproximações para os correspondentes vectores próprios ortonormados. Assim, no método de Jacobi, uma base ortonormada de vectores próprios é calculada simultaneamente com os valores próprios, sendo esta uma das razões que poderão influir na escolha da utilização deste método. Apesar de existirem métodos mais eficientes de cálculo dos valores próprios de uma matriz, o método de Jacobi é ainda bastante popular devido à sua simplicidade, facilidade de implementação e estabilidade numérica.

## 4.2 Determinação de valores próprios de matrizes tridiagonais – método da sequência de Sturm

Como vimos na Secção 2.3, os métodos de Givens e de Householder são usados para reduzir uma matriz real simétrica (ou uma matriz hermitiana) à forma tridiagonal simétrica. Vejamos agora como resolver o problema da determinação dos valores próprios de uma matriz desta forma.

O método que descrevemos nesta secção destina-se a obter informação acerca da *localização* de um ou alguns valores próprios de  $A$  (tridiagonal, simétrica, real) podendo, no entanto, esta técnica ser utilizada para calcular uma aproximação para qualquer valor próprio particular. Se estivermos interessados no cálculo de todos os valores próprios de  $A$  e correspondentes vectores próprios esta não será a forma mais eficiente para a sua determinação. Um algoritmo especialmente indicado para esse caso será discutido posteriormente.

Seja então  $A$  uma matriz real, simétrica e tridiagonal,

$$A = \begin{pmatrix} a_1 & b_2 & & \\ b_2 & a_2 & b_3 & \\ & & \dots & \\ & & & b_{n-1} & a_{n-1} & b_n \\ & & & & b_n & a_n \end{pmatrix} \quad (4.8)$$

Podemos assumir que  $A$  é *irredutível* i.e., que  $b_i \neq 0$ ;  $i = 2, \dots, n$ . Com efeito, se, para algum  $i$ ,  $b_i = 0$ , então  $\det(A - \lambda I)$  poder-se-ia exprimir como produto de dois determinantes de matrizes tridiagonais de menores dimensões, e os resultados aplicar-se-iam a cada um deles.

Seja

$$p_i(\lambda) = \det(A_i - \lambda I_i); \quad i = 1, 2, \dots, n,$$

onde  $A_i$  é a submatriz principal de  $A$  formada pelas suas primeiras  $i$  linhas e  $i$  colunas, i.e.

$$A_i = \begin{pmatrix} a_1 - \lambda & b_2 & & \\ b_2 & a_2 - \lambda & b_3 & \\ & & \dots & \\ & & & b_{i-1} & a_{i-1} - \lambda & b_i \\ & & & & b_i & a_i - \lambda \end{pmatrix}. \quad (4.9)$$

Temos então,

$$p_1(\lambda) = a_1 - \lambda$$

$$p_2(\lambda) = (a_2 - \lambda)(a_1 - \lambda) - b_2^2,$$

e, em geral, expandindo  $p_i(\lambda)$  ao longo da sua última coluna,

$$p_i(\lambda) = (a_i - \lambda)p_{i-1}(\lambda) - b_i^2 p_{i-2}(\lambda); \quad i = 3, \dots, n.$$

Definindo  $p_0(\lambda) := 1$ , obtemos assim a seguinte relação de recorrência:

$$\left. \begin{aligned} p_0(\lambda) &= 1, \\ p_1(\lambda) &= a_1 - \lambda, \\ p_i(\lambda) &= (a_i - \lambda)p_{i-1}(\lambda) - b_i^2 p_{i-2}(\lambda); \quad i = 2, \dots, n. \end{aligned} \right\} \quad (4.10)$$



**Definição 4.1** Seja  $\{p_0, p_1, \dots, p_n\}$  uma sequência finita de polinómios reais. Esta sequência diz-se uma **sequência de Sturm num intervalo**  $(a, b)$  (onde  $a$  ou  $b$  podem ser infinitos) se verificar as seguintes condições :

- $p_0(\lambda)$  não se anula em  $(a, b)$ .
- Se  $p_i(\xi) = 0$ ;  $1 \leq i \leq n-1, a \leq \xi \leq b$ , então

$$p_{i-1}(\xi)p_{i+1}(\xi) < 0.$$

**Lema 4.1** A sequência de polinómios definida por (4.10) (onde  $b_i \neq 0$ ;  $i = 2, \dots, n$ ) é uma sequência de Sturm.

**Dem:** Da relação de recorrência, vem

$$p_{i+1}(\lambda) = (a_{i+1} - \lambda)p_i(\lambda) - b_{i+1}^2 p_{i-1}(\lambda); \quad i = 1, \dots, n-1,$$

onde  $p_0(\lambda) = 1$ ,  $p_1(\lambda) = a_1 - \lambda$  e  $b_{i+1} \neq 0$ . Então, se  $p_i(\xi) = 0$ , temos

$$p_{i+1}(\xi) = -b_{i+1}^2 p_{i-1}(\xi),$$

ou seja,

$$p_{i+1}(\xi)p_{i-1}(\xi) \leq 0. \tag{4.11}$$

Uma vez que  $b_{i+1}^2 > 0$ , a igualdade em (4.11) só pode ocorrer se

$$p_{i+1}(\xi) = p_{i-1}(\xi) (= p_i(\xi)) = 0.$$

Mas, nesse caso, ter-se-á necessariamente, por causa da relação de recorrência,

$$p_{i-2}(\xi) = p_{i-3}(\xi) = \dots = p_0(\xi) = 0,$$

o que contradiz o facto de  $p_0(\lambda) \equiv 1$ .

Conclusão:

$$p_{i+1}(\xi)p_{i-1}(\xi) < 0.$$

□

**Lema 4.2** Os zeros do polinómio  $p_{i+1}$  são distintos e estritamente separados pelos zeros de  $p_i$ ;  $i = 1, \dots, n-1$ .

(Nota: Os zeros de  $p_i$ ;  $i = 1, 2, \dots, n$ , são reais.)

**Dem:** (Por indução sobre  $i$ )

1<sup>o</sup>) Facilmente se verifica que o zero de  $p_1$  separa os dois zeros de  $p_2$ , ou seja, que as raízes  $\xi_1$  e  $\xi_2$  de  $p_2$  são tais que  $\xi_1 < a_1$  e  $\xi_2 > a_1$ .

2<sup>o</sup>) Assumamos, por hipótese de indução, que os zeros de  $p_i$  são distintos e estritamente separados pelos zeros de  $p_{i-1}$  ( $i \geq 2$ ). Então,  $p_{i-1}$  muda de sinal entre dois zeros consecutivos de  $p_i$ . Mas, em cada zero  $\xi$  de  $p_i$ ,  $p_{i-1}(\xi)p_{i+1}(\xi) < 0$ , logo  $p_{i+1}$  também muda de sinal entre cada dois zeros adjacentes de  $p_i$ , ou seja,  $p_{i+1}$  tem  $(i - 1)$  zeros colocados estritamente entre os  $i$  zeros de  $p_i$ . Mas,

$$\lim_{\lambda \rightarrow -\infty} p_{i+1}(\lambda) = \lim_{\lambda \rightarrow -\infty} p_{i-1}(\lambda)$$

e

$$\lim_{\lambda \rightarrow +\infty} p_{i+1}(\lambda) = \lim_{\lambda \rightarrow +\infty} p_{i-1}(\lambda).$$

(Note-se que  $p_i(\lambda) = (-\lambda)^i + \text{pol. de grau } (i - 1) \text{ em } \lambda$ .) Logo, para  $\lambda$  suficientemente pequeno ou  $\lambda$  suficientemente grande,  $p_{i-1}(\lambda)$  e  $p_{i+1}(\lambda)$  têm o mesmo sinal, o que significa que  $p_{i+1}$  muda de sinal (i.e., tem um zero) à esquerda do menor zero de  $p_i$  e tem um zero à direita do maior zero de  $p_i$ . Como  $p_{i+1}$  tem  $(i + 1)$  zeros, conclui-se que os  $i$  zeros de  $p_i$  separam estritamente os  $i + 1$  zeros de  $p_{i+1}$ .

□

Como consequência imediata do lema anterior, obtemos o seguinte

**Corolário 4.1** *Os valores próprios de  $A$  (i.e., os zeros de  $p_n(\lambda)$ ) são distintos e, portanto,  $A$  é uma matriz não-deficiente.*

Com base nos lemas anteriores, podemos agora demonstrar o seguinte

**Teorema 4.2 (Propriedade da sequência de Sturm)** *Dado  $x$ , seja  $\nu(x)$  o número de mudanças de sinal na sequência numérica  $\{p_0(x), p_1(x), \dots, p_n(x)\}$  onde, se  $p_i(x) = 0$ , o seu sinal é definido como contrário ao de  $p_{i-1}(x)$ . Então,  $\nu(x)$  é igual ao número de valores próprios de  $A$  no intervalo  $(-\infty, x]$ .*

**Dem:** Considere-se o comportamento de  $\nu(x)$  quando  $x$  varia.

(i) Para  $x$  suficientemente pequeno,  $p_i(x) \approx (-x)^i > 0, \forall i$ , ou seja,  $\nu(x) = 0$ .

Como os polinômios  $p_0, p_1, \dots, p_n$  são funções contínuas, segue-se que  $\nu(x)$  só pode mudar em pontos  $x$  que sejam zeros de algum polinômio  $p_i$ .

(ii) Vejamos primeiramente que  $\nu(x)$  permanece constante em cada zero de  $p_i$ ;  $i = 1, 2, \dots, n-1$  :

Seja  $\xi$  tal que, para um certo  $i$ ,  $1 \leq i \leq n-1$ , se tem  $p_i(\xi) = 0$ . Como os zeros de  $p_i$  são estritamente separados pelos zeros de  $p_{i-1}$  e  $\lim_{x \rightarrow -\infty} \text{sgn}(p_i(x)) = \lim_{x \rightarrow -\infty} \text{sgn}(p_{i-1}(x))$  facilmente se conclui que

$$\text{sgn}(p_{i-1}(\xi - h)) = \text{sgn}(p_i(\xi - h)),$$

para  $h > 0$ , suficientemente pequeno. Além disso, atendendo a que  $p_{i-1}(\xi)p_{i+1}(\xi) < 0$ , segue-se, por continuidade que nas vizinhanças de  $\xi$  os sinais dos polinômios  $p_{i-1}$ ,  $p_i$  e  $p_{i+1}$  se comportarão como se ilustra numa das duas tabelas seguintes:

	$\xi - h$	$\xi$	$\xi + h$		$\xi - h$	$\xi$	$\xi + h$
$i - 1$	-	-	-	$i - 1$	+	+	+
$i$	-	0	+	$i$	+	0	-
$i + 1$	+	+	+	$i + 1$	-	-	-

Em ambos os casos, tem-se  $\nu(\xi - h) = \nu(\xi) = \nu(\xi + h)$ , logo o número de mudanças de sinal permanece constante.

(iii) Resta-nos mostrar que  $\nu(x)$  aumenta de uma unidade sempre que  $x$  "passa" por um zero de  $p_n$ , ou seja, por um valor próprio de  $A$ :

Seja então  $\xi$  um zero de  $p_n$ . Neste caso, facilmente se verifica que, na vizinhança de  $\xi$  existem apenas duas possibilidades para a sequência de sinais, ilustradas nas tabelas seguintes:

	$\xi - h$	$\xi$	$\xi + h$		$\xi - h$	$\xi$	$\xi + h$
$n - 1$	+	+	+	$n - 1$	-	-	-
$n$	+	0	-	$n$	-	0	+

Em ambos os casos  $\nu(\xi)$  aumenta de uma unidade, o que completa a demonstração.

□

Uma procura sistemática, usando o Teorema 4.2 para diversos valores de  $x$ , permite-nos localizar intervalos contendo cada um dos valores próprios de  $A$ . Com efeito, se

$$\nu(a) = k \quad \text{e} \quad \nu(b) = k + 1, \quad (a < b),$$

então existe um e um só valor próprio de  $A$  no intervalo  $(a, b]$ . Depois de encontrado um intervalo contendo exactamente um valor próprio de  $A$ , o método da bissecção de intervalos, juntamente com a propriedade da sequência de Sturm, permite-nos obter uma estimativa para esse valor próprio. Mais precisamente, se  $\nu(a) = k$  e  $\nu(b) = k + 1$ , seja

$$\mu = \frac{1}{2}(a + b).$$

Então,  $\nu(\mu) = k$  ou  $\nu(\mu) = k + 1$ . Se  $\nu(\mu) = k$  o valor próprio  $\lambda_{k+1}$  pertence ao intervalo  $(\mu, b]$ ; caso contrário,  $\lambda_{k+1}$  pertence ao intervalo  $(a, \mu]$ . Este processo pode ser repetido para determinar um intervalo de amplitude muito pequena no qual se encontre o valor próprio  $\lambda_{k+1}$  de  $A$ , ou seja, para obter uma aproximação para  $\lambda_{k+1}$ .

Com aritmética exacta, o método acima referido poderia ser usado para determinar qualquer valor próprio de  $A$  com a precisão desejada. Na prática, tal como sabemos, os erros de arredondamento limitam a precisão obtida. Contudo, pode provar-se que este é um processo excepcionalmente estável, mesmo quando os valores próprios de  $A$  são bastante próximos uns dos outros; veja, por exemplo, [Wil65, pp 303-306].

Terminamos esta secção referindo que a determinação dos vectores próprios associados aos valores próprios obtidos poderá ser feita usando o método da potência inverso, descrito na Secção 3.4. A decomposição  $LU$  de  $A - \lambda I$  será bastante eficiente se tirarmos partido da forma tridiagonal dessa matriz.

### 4.3 Método QR Básico

A maioria dos algoritmos existentes hoje em dia, para a resolução de problemas de valores próprios de uma matriz  $A$ , passa pela determinação da factorização de Schur de  $A$ . A decomposição de Schur de  $A$ ,

$$A = Q T Q^*,$$

onde  $Q$  é uma matriz unitária e  $T$  é uma matriz triangular superior, pode ser obtida transformando  $A$ , através de uma sequência de transformações de semelhança unitárias

$$X \longrightarrow Q_j^* X Q_j,$$

de modo que

$$\underbrace{Q_j^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_j}_{Q},$$

converge para uma matriz triangular inferior  $T$ , quando  $j \rightarrow \infty$ .

A obtenção desta decomposição é geralmente dividida em duas fases. Na primeira fase, no caso da matriz  $A$  ser não hermitiana, um método directo é aplicado para produzir uma matriz Hessenberg  $H$ . Na segunda fase é usado um método iterativo para gerar uma sequência “infinita” de matrizes Hessenberg que converge para a forma triangular. (Relembre que todo o processo para obtenção dos valores próprios de uma matriz tem que ser iterativo). Esquemáticamente, o processo pode ser descrito do seguinte modo:

$$\begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{pmatrix} \xrightarrow{\text{Fase 1}} \begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{pmatrix} \xrightarrow{\text{Fase 2}} \begin{pmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{pmatrix}$$

$A \neq A^*$   $H$   $T$

Se  $A$  é uma matriz de ordem  $n$ , a redução de  $A$  à forma Hessenberg da Fase 1 requer  $O(n^3)$  flops. Em geral, o método iterativo da Fase 2, converge (com a precisão da máquina) em  $O(n)$  iterações. Como cada iteração envolve  $O(n^2)$  flops, o esforço computacional total será de  $O(n^3)$  flops. Estes valores explicam a importância da Fase 1. Sem esta fase preliminar, cada iteração da Fase 2 envolveria a matriz toda, requerendo assim  $O(n^3)$  flops o que conduziria a um esforço total de, pelo menos,  $O(n^4)$  flops.

Se a matriz  $A$  for hermitiana, então a matriz  $Q_j^* \cdots Q_2^* Q_1^* A Q_1 Q_2 \cdots Q_j$  também é hermitiana. No limite, a matriz que se obtém é simultaneamente triangular e hermitiana, ou seja é diagonal. Neste caso, na Fase 1 a matriz intermédia é agora uma matriz hermitiana com a forma Hessenberg, isto é, é uma matriz tridiagonal. Tal como foi referido anteriormente, a matriz que se obtém na Fase 2 é uma matriz triangular hermitiana, ou seja é uma matriz diagonal. Esquemáticamente, temos agora:

$$\begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{pmatrix} \xrightarrow{\text{Fase 1}} \begin{pmatrix} x & x & & & \\ x & x & x & & \\ & x & x & x & \\ & & x & x & x \\ & & & x & x \end{pmatrix} \xrightarrow{\text{Fase 2}} \begin{pmatrix} x & & & & \\ & x & & & \\ & & x & & \\ & & & x & \\ & & & & x \end{pmatrix}$$

$A = A^*$   $T$   $D$

Nesta secção descrevemos um algoritmo, conhecido como *método QR* e que consiste numa sequência de transformações de semelhança aplicadas a uma matriz  $A_1 \equiv A$  que, no limite, reduzem esta matriz à forma triangular superior ou diagonal. Na prática, podemos assumir que  $A_1$  é já uma matriz tridiagonal ou Hessenberg superior, formas às quais, como já vimos, é sempre possível reduzir uma matriz dada inicialmente. O algoritmo QR, desenvolvido em 1961 por Francis [Fra62] e por [Kub61] independentemente, baseia-se no uso de transformações de semelhança unitárias. Em muitos aspectos, este é provavelmente um dos processos mais eficientes para a resolução do problema da determinação dos valores próprios e vectores próprios de uma matriz.

Dada uma matriz arbitrária  $A \in M_n$ , existe sempre uma matriz unitária  $Q^*$  tal que

$$Q^* A = R, \quad (4.12)$$

com  $R$  triangular superior. Tal matriz  $Q^*$  pode ser construída de forma estável como produto de matrizes de rotação plana ou de matrizes hermitianas elementares — reveja, a este propósito, a Secção 2.2, onde é discutida a utilização de matrizes desse tipo para a redução à forma  $(k \ 0 \ \dots \ 0)^T$  de um vector  $a = (a_1 \ a_2 \ \dots \ a_n)^T$ . Isto significa que qualquer matriz admite uma decomposição na forma

$$A = QR, \quad (4.13)$$

com  $Q$  unitária e  $R$  triangular superior. Facilmente se prova também que, se  $A$  é não-singular e  $R$  tiver elementos diagonais positivos, então a decomposição (4.13) é única; veja, por exemplo, [Wil65, pp 241 e segs]. Além disso, se  $A$  é real,  $Q$  e  $R$  são também matrizes reais.

O método QR é baseado na decomposição QR de uma matriz. Suponhamos que  $A_1 \equiv A$  e consideremos a decomposição de  $A_1$  na forma

$$A_1 = Q_1 R_1,$$

onde  $Q_1$  é uma matriz unitária e  $R_1$  é uma matriz triangular superior. Seja

$$A_2 = R_1 Q_1,$$

i.e.,  $A_2$  é obtida invertendo a ordem das matrizes  $Q_1$  e  $R_1$ . Então,

$$\begin{aligned} A_2 &= Q_1^{-1} Q_1 R_1 Q_1 \\ &= Q_1^* A_1 Q_1, \end{aligned}$$

ou seja,  $A_2$  é unitariamente semelhante a  $A_1$ . Aplicando esta estratégia iterativamente, obtemos o seguinte algoritmo, que constitui o chamado *Método QR básico*:

**Método QR básico:**

$$\left. \begin{aligned} A_1 &\equiv A, \\ Q_k R_k &= A_k \\ A_{k+1} &= R_k Q_k \end{aligned} \right\} k = 1, 2, \dots, \quad (4.14)$$

onde  $Q_k R_k$  é a decomposição QR de  $A_k$ , ou seja,  $Q_k$  é unitária e  $R_k$  é triangular superior de elementos diagonais reais positivos.

Facilmente se verifica que o método QR é um método envolvendo sucessivas transformações de semelhança. Mais precisamente, tem-se

**Lema 4.3** *Seja  $\{A_k\}$  a sequência de matrizes definida por (4.14). Então:*

$$(i) \quad A_{k+1} = (Q_1 \dots Q_k)^* A_1 (Q_1 \dots Q_k).$$

$$(ii) \quad \text{Sendo } Q_k := Q_1 \dots Q_k \text{ e } R_k := R_k \dots R_1, \text{ tem-se}$$

$$A^k = Q_k R_k.$$

**Dem:** Ao cuidado dos alunos.

□

As condições de convergência do método QR são apresentadas no teorema seguinte.

**Teorema 4.3** *Seja  $A_1 := A \in M_n$  uma matriz que verifica as seguintes condições :*

1. *Os valores próprios  $\lambda_i$  de  $A$  têm módulos distintos,*

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|.$$

2. *A matriz  $Y := X^{-1}$  onde  $X$  é a matriz cujas colunas são os vectores próprios de  $A$  admite uma decomposição LU*

$$Y = L_Y U_Y.$$

*Então, as matrizes  $A_k, Q_k, U_k$  do método (4.14) têm as seguintes propriedades de convergência:*

Existem matrizes diagonais  $D_k = \text{diag}(d_1^{(k)}, d_2^{(k)}, \dots, d_n^{(k)})$  cujos elementos  $d_i^{(k)}$  têm módulo 1 e tais que

$$\lim_{k \rightarrow \infty} D_{k-1}^* Q_k D_k = I, \quad (4.15)$$

e

$$\lim_{k \rightarrow \infty} D_{k-1}^* R_k D_k = \lim_{k \rightarrow \infty} D_{k-1}^* A_k D_{k-1} = \begin{pmatrix} \lambda_1 & * & \dots & * \\ & \lambda_2 & \dots & * \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}. \quad (4.16)$$

Em particular,

$$\lim_{k \rightarrow \infty} a_{jj}^{(k)} = \lambda_j, \quad j = 1, \dots, n. \quad (4.17)$$

**Dem.** Veja, por exemplo, [Wil65, pp 517 e segs].

□

O tipo de convergência expresso por (4.15) e (4.16) acima costuma referir-se por *convergência essencial*, isto é, convergência a menos do produto por certos elementos de módulo 1. Assim, o método QR, sob as condições do teorema anterior, converge *essencialmente* para uma matriz triangular com os valores próprios de  $A$  dispostos por ordem decrescente de grandeza ao longo da diagonal.

**Nota 4.1** A convergência para a forma triangular é assegurada, mesmo que  $Y$  não admita uma decomposição LU. A existência de  $L_Y$  e  $U_Y$  garante apenas que os valores próprios de  $A$  aparecem ao longo da diagonal por ordem decrescente de grandeza.

**Nota 4.2** A convergência de  $A_k$  para a forma triangular é linear, sendo tanto mais rápida quanto menores forem as razões  $|\lambda_i/\lambda_j|$ ,  $i > j$ , i.e., quanto maior for a separação dos valores próprios. Mais concretamente, prova-se que, sendo  $A_k = (a_{ij}^{(k)})$  se tem

$$a_{ij}^{(k)} = O(|\frac{\lambda_i}{\lambda_j}|^k), \quad k \rightarrow \infty, \quad i > j; \quad (4.18)$$

**Nota 4.3** Facilmente se prova que o método QR preserva a forma Hessenberg superior. Como o método preserva também a simetria, uma vez que envolve transformações de semelhança unitárias, segue-se que, se  $A$  é simétrica tridiagonal, o método QR preservará essa forma. Assim, e com o objectivo de reduzir o número de operações por iteração, deverá sempre reduzir-se a matriz inicial à forma Hessenberg superior, ou, caso seja possível, à forma tridiagonal simétrica. Tal estratégia reduz, tal como foi referido anteriormente, o número de operações por iteração de  $O(n^3)$  para  $O(n^2)$ .



## 4.4 Método QR com translação de origem

A convergência para zero dos elementos abaixo da diagonal pode ser melhorada significativamente com introdução no algoritmo de translações de origem; reveja a Secção 3.2. Suponhamos que a matriz  $A_1$  e, conseqüentemente todas as matrizes  $A_k$ , têm a forma Hessenberg superior e consideremos a aplicação do método QR para a matriz  $\tilde{A}_1 = A_1 - pI$ . Admitindo que os valores próprios de  $A_1$  são tais que

$$|\lambda_1 - p| > \dots > |\lambda_n - p|,$$

então a convergência do único elemento não-nulo  $\tilde{a}_{n,n-1}^{(k)}$  na última linha de  $\tilde{A}_k$  para zero comportar-se-á como

$$\left(\frac{\lambda_n - p}{\lambda_{n-1} - p}\right)^k.$$

Se  $p$  é uma boa aproximação para  $\lambda_n$ , por forma que  $|\lambda_n - p| \ll |\lambda_{n-1} - p|$ , então a convergência de  $\tilde{a}_{n,n-1}^{(k)}$  para zero será bastante rápida.

Assim, a convergência do método pode ser acelerada, se incorporarmos no algoritmo translações de origem adequadas, i.e. se o método QR básico for substituído pela sua seguinte extensão, conhecida como *Método QR com translação de origem*.

**Método QR com translação de origem:**

$$\left. \begin{aligned} A_1 &\equiv A, \\ A_k - p_k I &= Q_k R_k \\ A_{k+1} &= R_k Q_k + p_k I \end{aligned} \right\} k = 1, 2, \dots \quad (4.19)$$

Note-se que

$$\begin{aligned} A_{k+1} &= Q_k^{-1} Q_k R_k Q_k + p_k I \\ &= Q_k^* (A_k - p_k I) Q_k + p_k I \\ &= Q_k^* A_k Q_k - p_k I + p_k I \\ &= Q_k^* A_k Q_k, \end{aligned}$$

ou seja, que  $A_{k+1}$  se obtém ainda de  $A_k$  por uma transformação de semelhança unitária.

Surge agora a questão de saber como determinar as translações  $p_k \approx \lambda_n$ . Começemos por supor que a matriz  $A$  é tridiagonal simétrica, pelo que os seus valores próprios são todos reais. Se os valores próprios de  $A$  têm módulos distintos, então é de esperar que  $a_{n,n-1}^{(k)}$  e  $a_{n,n}^{(k)}$  tendam para zero e  $\lambda_n$ , respectivamente. É assim natural tomar  $p_k = a_{n,n}^{(k)}$  desde que  $a_{n,n-1}^{(k)}$  seja "pequeno" ou  $a_{n,n}^{(k)}$  mostre sinais de convergência, por exemplo, desde que

$$\left| 1 - \frac{a_{n,n}^{(k-1)}}{a_{n,n}^{(k)}} \right| \leq \epsilon < 1.$$

**Nota 4.4** Em [TB97], mostra-se que esta escolha da translação de origem está implicitamente relacionada com o quociente de Rayleigh correspondente ao vector próprio aproximado pela última coluna de  $Q_k$ . Por esta razão, esta translação é conhecida como *translação do quociente de Rayleigh*.

O método QR com translação de origem do quociente de Rayleigh tem, sob certas condições, convergência *localmente* cúbica; veja [Wil65, p.548]. No entanto, a convergência não está garantida em todas as situações (veja Exercício 10).

Outra escolha possível da translação é definida do seguinte modo. Seja  $S_k$  a submatriz  $2 \times 2$  contida no canto inferior direito de  $A_k$  i.e.

$$S_k := \begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{pmatrix}.$$

A *translação de Wilkinson* é definida como sendo o valor próprio de  $S_k$  mais próximo de  $a_{n,n}^{(k)}$ . (Note-se que esta matriz é simétrica, uma vez que  $A_k$  é tridiagonal simétrica). No caso de os dois valores próprios de  $S_k$  estarem a igual distância de  $a_{n,n}^{(k)}$ , escolhe-se um deles (arbitrariamente) para definir a translação.

O método QR com translação de Wilkinson tem também, em geral, convergência cúbica. Além disso, pode provar-se que, no pior dos casos, essa convergência é quadrática. Assim sendo, conclui-se que o método QR com translação de Wilkinson converge sempre (em aritmética exacta).

Se  $A$  é real, mas tem valores próprios complexos conjugados, i.e. se  $\lambda_{n-1} = \overline{\lambda_n}$ , então é evidente que  $a_{n,n}^{(k)}$ , não sendo uma aproximação para  $\lambda_n$ , não será uma boa escolha para  $p_k$ . Neste caso, determinam-se os valores próprios da submatriz  $S_k$ , definida anteriormente e efectuem-se duas translações de origem com esses valores próprios.

Para mais pormenores sobre o modo como podem ser escolhidos os valores de  $p_k$ , por forma a acelerar a convergência do método QR, veja, [Wil65, pp 507-509, 513]. Podemos resumir aqui as conclusões de Wilkinson [Wil65] :

1. Se  $A$  é real e tem valores próprios reais , então:

(a) Em cada passo  $k$  toma-se  $p_k := a_{n,n}^{(k)}$ ,

ou

(b) Determinam-se os valores próprios da submatriz  $S_k$  e:

- se eles forem reais, toma-se para  $p_k$  o mais próximo de  $a_{n,n}^{(k)}$ .
- se eles forem complexos conjugados,  $u_k \pm iv_k$ , digamos, toma-se  $p_k = u_k$ .

2. Se  $A$  é complexa, determinam-se os valores próprios de  $S_k$  e toma-se para  $p_k$  o mais próximo de  $a_{n,n}^{(k)}$ .

3. Se  $A$  é real, mas com possíveis valores próprios complexos, determinam-se os valores próprios de  $S_k$  e utilizam-se duas translações de origem com esses valores próprios (reais ou complexos conjugados).

## DEFLAÇÃO

Com translações de origem adequadas, ter-se-á  $a_{n-2,n-1}^{(k)} a_{n,n-1}^{(k)} \rightarrow 0$  rapidamente quando  $k \rightarrow \infty$ . Quando  $a_{n,n-1}^{(k)}$  for negligível, dentro da precisão com que estamos a trabalhar, podemos tomá-lo como zero, sendo nesse caso  $a_{n,n}^{(k)}$  tomado como um valor próprio de  $A$ . Os restantes valores próprios de  $A$  são então os valores próprios da submatriz principal contida nas primeiras  $(n-1)$  linhas e  $(n-1)$  colunas, pelo que poderemos agora prosseguir com a aplicação do método  $LR$  para essa matriz de ordem inferior à inicial. Continuando desta forma, podemos encontrar os valores próprios de  $A$ , um a um, trabalhando com matrizes de ordem cada vez menor. Se  $a_{n-1,n-2}^{(k)}$  for negligível e  $a_{n,n-1}^{(k)}$  significativo, podemos determinar os dois valores próprios da matriz  $S_k$ , considerá-los como valores próprios de  $A$  e fazer deflação omitindo as duas últimas linhas e colunas de  $A_k$ .

## CONCLUSÃO

O método QR com escolha apropriada da translação de origem tem sido, desde os anos 60, o método *standard* para calcular todos os valores próprios de uma matriz. Apresentamos, de

seguida, uma versão deste algoritmo, a qual tem incorporadas três modificações que já referimos e que tornam este algoritmo verdadeiramente eficiente, a saber:

1. Redução da matriz inicial  $A$  à forma Hessenberg ou tridiagonal (Fase 1).
2. Utilização de translações de origem apropriadas.
3. Deflação.

#### Método QR com translação de origem II

**I.**  $Q_0^* A_0 Q_0 = A$  %  $A_0$  matriz Hessenberg/tridiagonal  
% unitariamente semelhante a  $A$ ;

**II.** Para  $k = 1, 2, \dots$

$\left\{ \begin{array}{l} \text{Escolher } p_k \\ Q_k R_k = A_{k-1} - p_k I \\ A_k = R_k Q_k + p_k I \end{array} \right.$	<p>% <math>p_k</math> translação origem apropriada;</p> <p>% factorização QR de <math>A_{k-1} - p_k I</math>;</p> <p>% iteração <math>k</math> do método QR;</p>
---	--

$\left\{ \begin{array}{l} \text{Se } A_k(j, j-1) \text{ for "suficientemente" pequeno, fazer } A_k(j, j-1) = 0 \text{ para obter} \\ \\ A_k = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \\ \\ \text{e aplicar o método QR a } A_1 \text{ e } A_2. \end{array} \right.$	<p>% Deflação.</p>
---	--------------------

## 4.5 Método QR duplo

Suponhamos que  $A_1$  é uma matriz real (não-simétrica) reduzida à forma Hessenberg superior e irreduzível, i.e tal que os elementos da subdiagonal são não-nulos. Consideremos dois passos sucessivos do método QR com translações de origem complexas conjugadas., i.e.

$$\left. \begin{array}{l} Q_k R_k = A_k - p_k I \\ A_{k+1} = R_k Q_k + p_k I \end{array} \right\} \text{ passo } k$$

$$\left. \begin{array}{l} Q_{k+1} R_{k+1} = A_{k+1} - \bar{p}_k I \\ A_{k+2} = R_{k+1} Q_{k+1} + \bar{p}_k I \end{array} \right\} \text{ passo } (k+1).$$

Note-se que

$$\begin{aligned} A_{k+1} &= Q_k^* Q_k R_K Q_k + p_k I \\ &= Q_k^* (A_k - p_k I) Q_k + p_k I \\ &= Q_k A_k Q_k. \end{aligned}$$

De modo análogo,

$$A_{k+2} = Q_{k+1}^* A_{k+1} Q_{k+1},$$

ou seja, temos

$$A_{k+2} = Q^* A_k Q, \quad (4.20)$$

com

$$Q := Q_k Q_{k+1}. \quad (4.21)$$

Mas,

$$\begin{aligned} Q_k Q_{k+1} R_{k+1} R_k &= Q_k (A_{k+1} - \bar{p}_k I) R_k \\ &= Q_k (Q_k^* A_k Q_k - \bar{p}_k I) R_k \\ &= (A_k - \bar{p}_k I) Q_k R_k \\ &= (A_k - \bar{p}_k I) (A_k - p_k I), \end{aligned} \quad (4.22)$$

i.e. temos

$$QR = B, \quad (4.23)$$

onde

$$Q := Q_k Q_{k+1}, \quad (4.24)$$

$$R := R_{k+1} R_k, \quad (4.25)$$

$$\begin{aligned} B &:= (A_k - \bar{p}_k I) (A_k - p_k I) \\ &= A_k^2 - (p_k + \bar{p}_k) A_k + p_k \bar{p}_k I \end{aligned} \quad (4.26)$$

As quantidades  $s := p_k + \bar{p}_k$  e  $t := p_k \bar{p}_k$  são reais e, portanto, a matriz  $B$  em (4.26) é também real. Além disso a matriz triangular superior  $R$  tem elementos diagonais positivos, uma vez que o mesmo se assume para  $R_k$  e  $R_{k+1}$ . Assim, a matriz  $Q$  na decomposição QR de  $B$  dada por (4.23) terá de ser também real (atendendo a que toda a matriz real admite uma decomposição QR com  $Q$  e  $R$  reais, sendo tal decomposição única se  $R$  tiver elementos diagonais positivos). Segue-se de (4.20) que  $A_{k+2}$  é também uma matriz real. A ideia do chamado *método QR duplo*, introduzido por Francis em [Fra62], é calcular  $A_{k+2}$  directamente de  $A_k$ , usando apenas aritmética real. Uma

forma de obter  $Q$  (e portanto,  $A_{k+2}$ ) directamente será efectuar a decomposição QR da matriz  $B$ , por exemplo, através da utilização de matrizes hermitianas elementares:

$$P_{n-2}P_{n-1} \dots P_1P_0B = R,$$

onde  $R$  é triangular superior e onde

$$P_i = \left( \begin{array}{c|c} I_i & \\ \hline & I_{n-i} - 2w_iw_i^T \end{array} \right)$$

é escolhida de modo a reduzir à forma  $(k \ 0 \ \dots \ 0)^T$  a coluna  $(i+1)$  da matriz  $P_{i-1} \dots P_0$ . Ter-se-á então

$$Q = P_0P_1 \dots P_{n-2}. \quad (4.27)$$

A desvantagem deste processo de determinação de  $Q$  é que ele envolve um número muito elevado de operações, nomeadamente com o cálculo da matriz  $B$  dada por (4.26). Francis [Fra62] descreveu um processo muito mais eficiente de calcular  $A_{k+2}$ , para o qual há necessidade de se calcular apenas a 1ª coluna de  $B$ . Tal processo baseia-se no seguinte

**Teorema 4.4** *Sejam  $H, A, Q \in M_n$  com  $Q$  unitária e  $H$  Hessenberg superior irredutível e de elementos subdiagonais positivos. Se*

$$H = Q^*AQ, \quad (4.28)$$

*então  $H$  e  $Q$  são determinadas univocamente pela 1ª coluna de  $Q$ .*

**Dem:** Veja, por exemplo, [Wil65, pp 351-353].

□

**Nota 4.5** A exigência, no teorema anterior, de que os elementos subdiagonais de  $H$  sejam positivos, é necessária para estabelecer completamente a unicidade da decomposição (4.28). De facto, desde que  $H$  seja irredutível, a decomposição  $H = Q^*AQ$  é determinada *essencialmente* pela primeira coluna de  $Q$ , isto é,  $H$  e  $Q$  são determinadas a menos do produto por elementos de módulo 1. No que se segue, ao aplicarmos o teorema, é a esta unicidade essencial que nos referimos.

Consideremos então o seguinte processo de determinação da matriz  $A_{k+2}$  dada por (4.20), directamente, usando apenas aritmética real, conhecido por *Método QR duplo*.

**Método QR duplo:**

1. Calcular  $b := 1^{\text{a}}$  coluna da matriz  $B = A_k^2 - sA_k + tI$ .

2. Calcular  $P_0 = I - 1w_0w_0^T$  tal que

$$P_0 b = k e_1 \quad (k > 0).$$

3. Calcular

$$C = P_0 A_k P_0.$$

4. Reduzir  $C$  à forma Hessenberg superior usando transformações de semelhança unitárias, por exemplo, por meio de matrizes hermitianas elementares:

$$\tilde{P}_{n-2} \dots \tilde{P}_1 (P_0 A_k P_0) \tilde{P}_1 \dots \tilde{P}_{n-2} = H.$$

5. Considerar  $H$  como  $A_{k+2}$ .

A demonstração de que a matriz  $H$  obtida pelo processo acima descrito coincide (essencialmente) com a matriz  $A_{k+2}$  é a seguinte:

Temos,

$$Q^* A Q = A_{k+2}$$

e

$$\tilde{Q}^* A \tilde{Q} = H,$$

com

$$Q = P_0 P_1 \dots P_{n-2}$$

e

$$\tilde{Q} = P_0 \tilde{P}_1 \dots \tilde{P}_{n-2}.$$

Mas, as matrizes unitárias  $Q$  e  $\tilde{Q}$  têm as respectivas primeiras colunas iguais (iguais à primeira coluna de  $P_0$ ) em virtude da forma particionada das matrizes  $P_i$  e  $\tilde{P}_i$ . Então, de acordo com o teorema,  $A_{k+2} = H$ .

## 4.6 Notas e referências adicionais

- O método de Jacobi data de 1846 [Jac46], mas foi "redescoberto" em 1949 por Goldstine, Murray e von Neumann [GMvN59]. Em [GMvN59], é feita uma análise de erros do

processo.

Um método de tipo Jacobi para matrizes mais gerais foi introduzido por Eberlein [Ebe62]. Ruhe [Ruh68] provou que tais métodos podem convergir quadraticamente. Programas para estes métodos foram descritos em [Ebe68] para matrizes reais e em [Ebe70] para matrizes complexas.

- A determinação dos valores próprios de uma matriz simétrica tridiagonal pelo método da sequência de Sturm foi descrita e analisada em [Giv54].
- O problema da determinação de vectores próprios de matrizes tridiagonais quando são conhecidas boas aproximações para os valores próprios foi considerado em [BS56], [For58], [Wil58] e [RLHK51].

## 4.7 Exercícios

1. Considere a seguinte matriz simétrica

$$A_1 = \begin{pmatrix} 10 & 3 & 0 \\ 3 & 3 & -2 \\ 0 & -2 & 1 \end{pmatrix}.$$

- a) Determine a matriz de rotação plana  $R_{1,2}(\theta)$  tal que o elemento na posição (1,2) de

$$A_2 = R_{1,2}(\theta) A_1 R_{1,2}(\theta)^T$$

é nulo.

- b) Calcule  $A_2$ .

(isto é, faça uma iteração do **método de Jacobi**).

2. Considere a seguinte matriz simétrica e tridiagonal

$$T = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{pmatrix}.$$

- a) Justifique, sem recorrer à função *eig* do *MATLAB*, que a matriz  $T$  tem valores próprios reais e distintos.
- b) Indique um intervalo que contenha todos os valores próprios  $\lambda_i$  de  $T$



- c) Consideremos os v.p.'s  $\lambda_i$  ordenados por ordem crescente,  $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$ . Use o **método das sequências de Sturm** para encontrar um intervalo que contenha  $\lambda_2$ , mas nenhum dos outros valores próprios.

3. Considere a matriz

$$T = \begin{pmatrix} 4 & 1 & & \\ 1 & 4 & 1 & \\ & 1 & 4 & 1 \\ & & 1 & 4 \end{pmatrix}.$$

- a) Quantos valores próprios de  $T$  são negativos?  
b) Quantos valores próprios de  $T$  existem no intervalo  $[5,6]$ ?

4. A *inércia* de uma matriz  $A$  de ordem  $n$  é definida pelo terno de inteiros

$$In(A) = (\pi(A), \nu(A), \delta(A)),$$

onde  $\pi(A)$ ,  $\nu(A)$  e  $\delta(A)$  designam o número de valores próprios de  $A$  com parte real positiva, negativa e nula, respectivamente.

- a) Mostre que, se

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix},$$

então

$$In(A) = (3, 1, 0).$$

- b) Sabendo que uma matriz  $B$  de ordem  $n$  é estável se

$$In(B) = (0, n, 0),$$

defina uma matriz  $B$  de ordem 4, tridiagonal irredutível, simétrica e estável.

5. Considere a matriz

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 2 & 2 & 3 & 0 \\ 0 & 3 & 3 & 4 \\ 0 & 0 & 4 & 4 \end{pmatrix}.$$

- a) Determine o número de valores próprios de  $A$ , no intervalo  $[8, 8.5]$ .
  - b) Determine um intervalo de amplitude  $10^{-1}$  que contenha o maior valor próprio de  $A$ .
6. Mostre que a redução prévia de uma matriz simétrica  $A \in \mathbb{R}^{n \times n}$  à forma tridiagonal representa, em geral, uma redução de  $O(n^4)$  para  $O(n^3)$  flops, na aplicação do método QR para obtenção dos valores próprios de  $A$ .
7. Seja  $A$  uma matriz real, simétrica e tridiagonal e considere a factorização QR dessa matriz, i.e.  $A = QR$ .
- a) Quais são, em geral, os elementos não nulos das matrizes  $R$  e  $Q$ ?
  - b) Verifique que a estrutura tridiagonal de  $A$  é recuperada, quando o produto  $RQ$  é formado.
  - c) Conclua que o método QR preserva a estrutura tridiagonal da matriz inicial.
8. Considere a matriz

$$T_1 = \begin{pmatrix} 10 & -3 & 0 & 0 \\ -3 & 4 & 0.5 & 0 \\ 0 & 0.5 & 6 & 0.1 \\ 0 & 0 & 0.5 & 1 \end{pmatrix}.$$

- a) Justifique que esta matriz é não deficiente.
- b) Obtenha a decomposição  $QR = T_1$ , onde  $Q$  é uma matriz ortogonal e  $R$  é uma matriz triangular superior.
- c) Calcule  $T_2 = RQ$ . As matrizes  $T_1$  e  $T_2$  são semelhantes? Justifique.
- d) Repita os cálculos anteriores com a matriz  $T - I$ . O que conclui?



9. Considere a matriz  $A = \text{hilb}(4)$ .

- a) Determine uma matriz tridiagonal  $T$ , unitariamente semelhante a  $A$ .
- b) Use o Método QR com translação de origem, juntamente com deflação para obter todos os valores próprios de  $A$ .

(Considere  $t_{m,m-1} = 0$ , sempre que  $|t_{m,m-1}| < 10^{-4}$ )



**10.** Considere a matriz

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

- a) Mostre que o método QR básico não converge, neste caso.
- b) Verifique que a escolha da translação do quociente de Rayleigh não tem qualquer efeito, também.
- c) Use o método QR com translação de origem de Wilkinson para determinar aproximações para os valores próprios de  $A$ .

# Bibliografia

- [And95] et al. E. Anderson, *LAPACK user's guide*, SIAM, Philadelphia, 1995.
- [BE78] R. H. Barlow and D. J. Evans, *A parallel organization of the bisection algorithm*, Comput. J. (1978), no. 22, 267–269.
- [BES83] R. H. Barlow, D. J. Evans, and J. Schanehchi, *Parallel multi-section applied to the eigenvalue problem*, Comput. J. (1983), no. 26, 6–9.
- [BG86] H. J. Bernstein and M. Goldstein, *Parallel implementation of bisection for the calculation of eigenvalues for tridiagonal matrices*, Computing (1986), no. 31, 85–91.
- [BH60] F. L. Bauer and A. S. Householder, *Moments and characteristic roots*, Numer. Math. (1960), no. 2, 42–53.
- [Bow84] G. s. et al Bowgen, *The solution of  $n$  linear equations on a  $p$ -processor parallel computer*, Report 2.30, DAP Support Unit, Queen Mary College, London (1984).
- [BS56] R. A. Brooker and F. H. Sumner, *The method of lanczos for calculating the characteristic roots and vectors of a real symmetric matrix*, Proc. Instn Elect. Engrs B (1956), no. 103 Sup 11., 114–119.
- [Cup81] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math. (1981), no. 36, 177–195.
- [Dem97] J.W. Demmel, *Applied numerical linear algebra*, SIAM, 1997.
- [Duf87] I. S. Duff, *The influence of vector and parallel processors on numerical analysis*, em *The State of the Art in Numerical Analysis*, Iserles, A. e Powell, M. J. D. Eds, Calrendon Press, Oxford, 1987.
- [Ebe62] P. J. Eberlein, *A Jacobi-type method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix*, SIAM J. Appl. Math. (1962), no. 10, 74–88.

- [Ebe68] J. Eberlein, P. J. and Boothroyd, *Solution to the eigenproblem by a norm reducing Jacobi type method*, Numer. Math. (1968), no. 11, 1–12.
- [Ebe70] P. J. Eberlein, *Solution of the complex eigenproblem by norm reducing Jacobi type method*, Numer. Math. (1970), no. 14, 232–245.
- [FH61] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc. (1961), no. 94, 1–23.
- [For58] G. E. Forsythe, *Singularity and near singularity in numerical analysis*, Amer. Math. Mon. (1958), no. 65, 229–240.
- [Fra62] J. G. F. Francis, *The QR transformation i-ii*, Computer J. (1961, 1962), no. 4, 265–271; 332–345.
- [Gan54] F.R. Gantmacher, *The theory of matrices*, Chelsea Publ., N. York., 1954.
- [Gar77] B. S. et al. Garbow, *Matrix eigensystem routines - eispack guide extension*, Springer-Verlag, New York, 1977.
- [Ger31] S. Gerschgorin, *Über die abgrenzung der eigenwerte einer matrix*, Izv. Akad. Nauk SSSR, Ser. fiz. mat. (1931), no. 6, 749–754.
- [Giv54] J. W. Givens, *Numerical computation of the characteristic values of a real symmetric matrix*, Oak. Ridge National Laboratory (1954), no. ORNL-1574.
- [GMvN59] H. H. Goldstein, F J. Murray, and J. von Neumann, *The Jacobi method for real symmetric matrices*, J. Ass. comp. Mach. (1959), no. 6, 59–96.
- [GVL97] G.H. Golub and C.F. Van Loan, *Matrix computations*, 3 ed., SIAM, 1997.
- [GW73] A. R. Gourlay and G. A. Watson, *Computational methods for matrix eigenproblems*, John Wiley, London, 1973.
- [H.71] Sameh A. H., *On Jacobi and Jacobi-type algorithms for a parallel machine*, Math. Comp. (1971), no. 25, 579–590.
- [Hig96] N.J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, 1996.
- [Hor85] C. R. Horn, R. A. e Johnson, *Matrix computations*, Cambridge Univ. Press, 1985.
- [Hou64] A.S. Householder, *The theory of matrices in numerical analysis*, Dover, N.York., 1964.

- [IJ86] I. C. F. Ipsen and E. Jessup, *The impact of parallel architectures on the solution of the eigenvalue problem*, Oak. Ridge National Laboratory (1986), no. Report YALEU/DCS/RR-444, Yale University.
- [Jac46] C. G. J. Jacobi, *The impact of parallel architectures on the solution of the eigenvalue problem*, Crelle's J. (1846), no. 30, 51–9.
- [JM84] Dongarra J. J. and C. B. Moler, *Eispack – a package for solving matrix eigenvalue problems, em sources and development of mathematical software*, Cowell, W. R. Ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [KG86] A. H. Karp and J. Greenstadt, *An Improved parallel Jacobi method for diagonalizing a symmetric matrix*, 1986, Report G320-3484, IBM Scientific Center, Palo Alto.
- [Kub61] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, USSR Comput. Math. Phys. (1961), no. 3, 637–657.
- [MD85] Pryce Modi, J. J. and J. D., *Efficient implementation of Jacobi's diagonalisation method on dap*, Numer. Math. (1985), no. 46, 443–454.
- [RLHK51] J. B. Rosser, C. Lanczos, M. R. Hestenes, and W. Karush, *The separation of close eigenvalues of a real symmetric matrix*, J. Res. Nat. Bur. Stand. (1951), no. 47, 291–297.
- [RR78] A. Ralston and P. Rabinowitz, *A first course in numerical analysis*, Mc Graw-Hill, Kogakusha, Tokyo, 1978.
- [Ruh68] A. Ruhe, *On the quadratic convergence of a generalization of the Jacobi method to arbitrary matrices*, BIT (1968), no. 8, 210–231.
- [SB80] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, Springer- Verlag, New York, 1980.
- [Sch61] A. Schonhage, Numer. Math. (1961), no. 3, 374–380.
- [Sch64] A. Schonhage, Numer. Math. (1964), no. 6, 410–412.
- [Smi76] B.T. et al. Smith, *Matrix eigensystem routines - eispack guide*, Springer- Verlag, New York, 1976.
- [Ste69] G. W. Stewart, *Accelerating the orthogonal iteration for the eigenvalues of a hermitian matrices*, Numer. Math. (1969), no. 13, 362–376.

- [Ste73] G. W. Stewart, *Introduction to matrix computations*, Academic Press, New York London, 1973.
- [Swa79] P. N. Swarztrauber, *A parallel algorithm for solving general tridiagonal equations*, Math. Comp. (1979), no. 33, 185–199.
- [Tau49] O. Taussky, *A recurring theorem for determinants*, Amer. Math. Monthly (1949), no. 56, 672–676.
- [TB97] L.N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, 1997.
- [Val90] M.R. Valença, *Métodos numéricos*, INIC, Braga, 1990.
- [Var65] R.S. Varga, *Minimal gerschgorin sets*, Pacific J. Math. (1965), no. 15, 719–729.
- [Wil58] J. H. Wilkinson, *The calculation of the eigenvectors of codiagonal matrices*, Computer J. I (1958), no. 8, 90–96.
- [Wil63] J. H. Wilkinson, *Rounding errors in algebraic processes*, Prentice-Hall, New Jersey, 1963.
- [Wil65] J. H. Wilkinson, *The algebraic eigenvalue problem*, Oxford Univ. Press, London, 1965.
- [WR71] J. H. Wilkinson and Reinsch, *Handbook for automatic*, C. Eds, 1971.